



## **Rapporto conclusivo**

### **Gruppo di lavoro “Codice sorgente aperto” (“Open Source”)**

*Versione 1.0*

## Sommario

Introduzione.....	4
Contenuti e scopo del documento.....	4
Il fenomeno OS: considerazioni di fondo.....	4
1.Scenario.....	6
1.1.Il settore ICT in Italia.....	6
1.2.Mercato ICT Italia.....	11
1.3.Il software open source: stato dell'arte.....	12
1.3.1.Analisi "Hype Cycle".....	12
1.3.2.Altre analisi.....	14
1.3.3.Uso del software OS nella pubblica amministrazione .....	16
1.3.4.Evoluzione tecnica e benchmark.....	18
1.3.5.Sintesi dello scenario del software OS.....	21
1.4.Iniziative pubbliche UE ed extra UE.....	22
1.4.1.Istituzioni UE.....	22
1.4.2.Paesi UE.....	23
1.4.3.Paesi extra UE.....	27
1.5.Il modello organizzativo e i processi produttivi dell'OS.....	29
1.5.1.Introduzione.....	29
1.5.2.Organizzazione dei team di progetto.....	30
1.5.3.Modello organizzativo: il caso Apache Software Foundation.....	30
1.5.4.Gestione delle risorse umane.....	31
1.5.5.Modello di sviluppo.....	31
1.5.6.Genesi del Progetto.....	32
1.5.7.Pianificazione di prodotto.....	33
1.5.8.Analisi dei requisiti e specifiche.....	33
1.5.9.Strumenti di sviluppo.....	34
1.5.10.Controllo di versione, system build e rilascio incrementale.....	34
1.5.11.Documentazione.....	35
1.5.12.Test.....	37
1.5.13.Supporto e manutenzione evolutiva.....	37
1.6.Scenario normativo.....	38
1.6.1.Il Diritto d'Autore e il software OS.....	38
1.6.2.Definizioni e aspetti giuridici del modello OS.....	40
1.7.Il modello open source e il mercato.....	44
1.7.1.I "prodotti" open source.....	45
1.7.2.I "prodotti accessori".....	46
1.7.3.I consulenti autonomi.....	46
1.7.4.Le società di assistenza tecnica.....	47
1.7.5.Le società di formazione e certificazione.....	47
1.7.6.Le "garanzie" sul software.....	47
2.Metodologia di attuazione della Direttiva del MIT.....	49
2.1.La Direttiva del Ministro Stanca.....	49
2.2.Impatto della Direttiva sulle procedure di acquisizione.....	50
2.2.1.Studio di fattibilità.....	52
2.2.2.Capitolato di gara.....	56
2.3.Metodologia di valutazione comparativa dei prodotti software.....	58
2.3.1.Individuazione dei prodotti OS "eleggibili".....	60
2.3.2.Individuazione dei prodotti proprietari candidati.....	62
2.3.3.Definizione pesi per la comparazione.....	63
2.3.4.Esclusione offerte con prodotti "non eleggibili".....	65
2.3.5.Assegnazione punteggi.....	66

2.3.6.Comparazione “semplice” .....	70
2.3.7.Il ruolo del CNIPA .....	72
2.4.Processo di valutazione nel caso di sviluppo ad hoc.....	72
3.Il Centro di Competenza.....	74
3.1.I Centri di Competenza in ambito UE.....	74
3.1.1.Austria.....	74
3.1.2.Francia.....	75
3.1.3.Germania.....	75
3.1.4.Olanda.....	75
3.2.Obiettivi del Centro di Competenza.....	75
3.3.Organizzazione del Centro di Competenza.....	76
4.Portale e Vetrina.....	78
4.1.Modello Logico della Vetrina.....	78
4.2.Disegno Funzionale.....	79
4.3.Mappa del sito.....	80
5.La piattaforma di sviluppo collaborativo.....	83
5.1.Modelli e strumenti per lo sviluppo collaborativo.....	83
5.2.Requisiti funzionali del CDE.....	83
5.3.Architettura.....	84
5.4.Strumenti tecnici innovativi.....	87

# Introduzione

## ***Contenuti e scopo del documento***

Il presente documento raccoglie una serie di informazioni relative al software "open source". A grandi linee, la trattazione riguarda:

- un aggiornamento dello scenario di riferimento rispetto alla situazione illustrata dalla Indagine conoscitiva del software Open Source (la cosiddetta "Commissione Meo");
- le iniziative UE e dei paesi extra UE;
- la descrizione del modello organizzativo e di sviluppo del software OS;
- l'attuale scenario normativo;
- il mercato rispetto al software OS;
- una metodologia di attuazione della Direttiva del Ministro per l'Innovazione e le Tecnologie del 19 dicembre 2003
- le caratteristiche di un costituendo Centro di Competenza nazionale in materia di OS per la Pubblica Amministrazione e delle sue principali infrastrutture e attività.

## ***Il fenomeno OS: considerazioni di fondo***

Nell'ambito dell'introduzione si ritiene utile esporre alcuni concetti generali, allo scopo di omogeneizzare sia la visione del fenomeno OS, sia il livello di comprensione e di interpretazione dei contenuti del presente documento.

Il software OS è nato spontaneamente con la diffusione di Internet nel mondo: è derivato dalla libertà totale della rete, dallo spazio illimitato alle iniziative singole o collettive, dalla possibilità di creare rapporti tra persone prescindendo da vincoli di spazio e tempo.

Il software, nella visione “filosofica” dei partecipanti al mondo OS (sviluppatori ed utenti), può assimilarsi a quella che ciascuno di noi ha dell'acqua, dell'aria, delle più elementari esigenze umane: il software viene visto come bene comune, gestito in modo totalmente trasparente, nella certezza che qualsiasi utile contributo porti vantaggio alla comunità umana.

L'affermazione crescente del software OS va di pari passo con la diffusione di Internet nel mondo: tra i primi benefici va sottolineata la conseguente, ugualmente progressiva diffusione di standard e formati aperti, ed i primi passi verso la stabilità, e quindi l'affidabilità dei prodotti OS attraverso la creazione delle community. Queste appaiono come un primo spontaneo, e necessario, assetto organizzativo condiviso per ricavare i benefici derivanti alla collettività dall'avere a disposizione il codice sorgente.

Da tale rilevazione delle caratteristiche, ad oggi, del fenomeno OS, scaturisce la sensazione che si è all'inizio di un processo di trasformazione della realtà del software, i cui risultati a lungo termine potrebbero modificare in maniera imprevedibile l'intero scenario.

Si osserva infatti che gli standard e i formati aperti, affermatasi con Internet, sono un primo passo verso una sorta di lingua comune che unisce di fatto gli utilizzatori della rete, e che non potrà che aggiornarsi e crescere nel tempo. Così come l'OS potrà essere fattore di

cambiamento, da una parte con la diffusione (in atto) della sua conoscenza e comprensione, dall'altra con iniziative di grandi dimensioni da parte di paesi in via di sviluppo, quali - ad esempio - la Cina.

Da quanto su espresso si ritiene che l'evoluzione dell'OS richieda un continuo aggiornamento della conoscenza, e la diffusione alle PA della stessa, stimolando il mercato perchè metta a disposizione adeguati servizi.

Nell'ambito del documento vengono illustrati e valutati i molteplici aspetti della problematica, con le conseguenti proposte per l'attuazione della Direttiva del Ministro.

# 1. Scenario

## 1.1. Il settore ICT in Italia

Per illustrare sinteticamente la situazione attuale dell'ICT italiano, e in particolare del settore della produzione del software, sembra opportuno partire dai dati e dall'analisi, ancora attuali, che l'indagine conoscitiva per il software a sorgente aperto nella PA (conosciuta come l'indagine della Commissione Meo) illustrava nel suo rapporto, e confrontarli con indagini più recenti svolte da altri addetti ai lavori<sup>1</sup>.

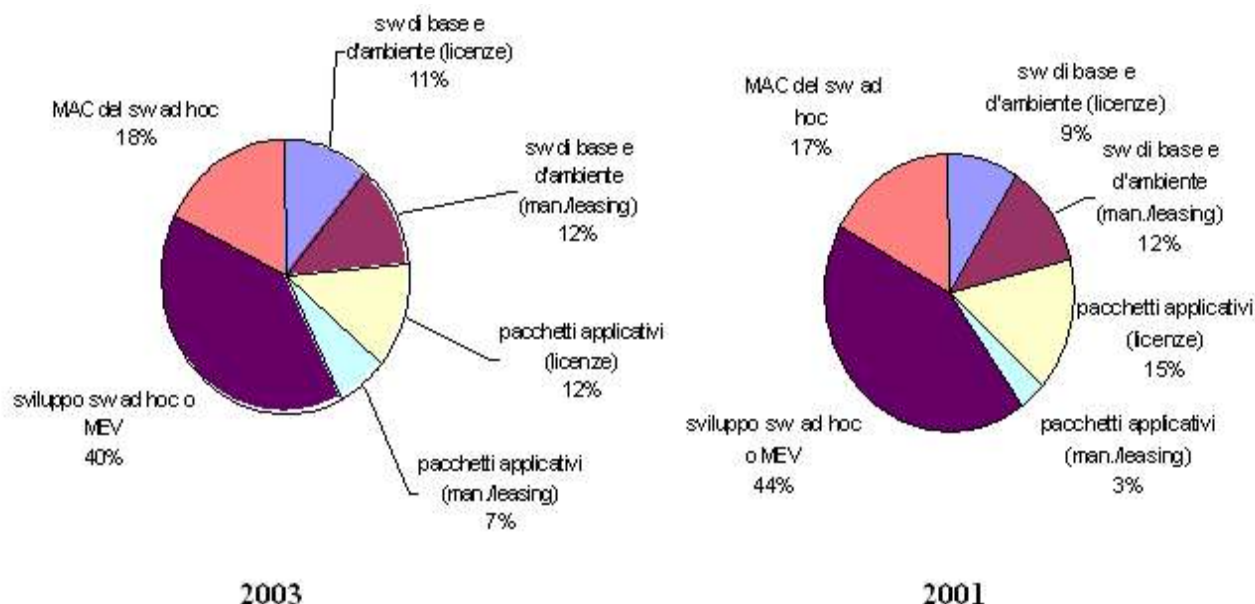
Ad esempio, secondo il rapporto Assinform 2004 e i dati di consuntivo del CNIPA, perdura il momento negativo del settore ICT nazionale. Il settore resta poco competitivo, e in forte sudditanza tecnologica rispetto ai corrispondenti settori di Paesi tecnologicamente avanzati.

La spesa delle Pubbliche amministrazioni italiane (sia centrali che locali), in software, nel 2003 si può stimare in 816 milioni di euro, in diminuzione del 6% rispetto all'anno precedente. Di questi, il 58% della spesa si è concentrata nello sviluppo, manutenzione e gestione di programmi custom, sviluppati cioè su commessa per una specifica amministrazione (o per più amministrazioni). Il rimanente 42% è stato utilizzato per l'acquisizione, manutenzione o leasing di licenze di pacchetti (sia di base che applicativi).

Più in dettaglio, la ripartizione dei costi attuali e precedenti è illustrata nei seguenti grafici a torta, dove:

- MAC = manutenzione migliorativa / adattiva / correttiva;
- MEV = manutenzione evolutiva.

**Figura 1.1: confronto spesa per il software nella P.A. italiana 2003-2001 (Fonte CNIPA)**



Rispetto ai dati riportati nel rapporto della Commissione Meo (che si riferivano all'anno 2001) è cresciuta l'incidenza percentuale della spesa relativa al software venduto su licenza

<sup>1</sup> Segnaliamo in particolare:

- 1° Rapporto sull'Open Source in Italia 2004 – Ware.it
- Analisi del fenomeno Open Source in Italia - ELISS

(dal 39% al 42%), mentre è diminuita la componente relativa al software ad hoc (dal 61% al 58%). Tuttavia, l'incidenza sul totale della spesa del software sviluppato ad hoc resta ancora maggioritaria. Ciò conferma e rafforza la necessità di considerare alternative al software proprietario venduto su licenza ed al software sviluppato ad hoc.

Da alcune indagini a carattere empirico (vista la carenza di dati in materia) svolte da soggetti interessati, si evince una crescente penetrazione del software open source (nel seguito: OS) nella pubblica amministrazione italiana, soprattutto in ambito locale.

Come dimostra la tabella 1.1 (fonte: rapporto Ware.it sul software OS in Italia), prendendo come riferimento due tra i componenti del mondo OS di ampia diffusione (il sistema operativo Linux e il web server Apache), la penetrazione del software OS è lenta ma in costante aumento. Il fenomeno si presenta in maniera più significativa nei comuni e nelle province, mentre riscontra maggior resistenza in ambito regionale.

**Tabella 1.1: Sistemi operativi e web server nella PA locale**

	Regioni		Province		Comuni capoluogo	
	2002	2003	2002	2003	2002	2003
<b>Sistemi operativi</b>						
	v.a.	v.a.	val. %	val. %	val. %	val. %
Microsoft	16	16	70,0	67,6	52,9	55,3
Sistemi operativi open source	2	3	23,0	28,4	37,3	37,9
Altri sistemi operativi	2	1	7,0	4,0	9,8	6,8
<b>Server web</b>						
Microsoft	15	16	68,0	62,7	47,1	49,5
Webserver Apache	1	3	27,0	30,4	41,2	41,7
Altro	4	1	7,0	6,9	11,8	8,7

Una seconda evidenza del crescente interesse verso le tematiche del software OS in ambito locale viene dall'aumento di iniziative ed eventi organizzati nelle varie regioni e province per promuovere la diffusione di conoscenza in materia. Tra tante citiamo, ritenendola la più completa e trasversale, la manifestazione organizzata dalla provincia di Pisa denominata SALPA<sup>2</sup>, che si svolge annualmente nella città toscana.

La crescente diffusione del software OS può essere messa in relazione diretta al numero crescente di associazioni e gruppi di sviluppatori OS (in particolare sviluppatori Linux) organizzati che si creano sul territorio nazionale. Come mostra la tabella successiva, in ogni regione (tranne il Molise) è presente una o più di queste entità; la distribuzione non è uniforme, concentrandosi in Lombardia, Veneto, Toscana ed Emilia Romagna.

<sup>2</sup> [www.salpa.pisa.it](http://www.salpa.pisa.it)

**Tabella 1.2: capoluoghi con associazioni e gruppi di sviluppatori Linux**

Regione	Numero comuni e città capoluogo coinvolte	Centri
Piemonte	4	Alessandria, Asti, Biella, Novara, Torino
Valle d'Aosta	1	Aosta
Lombardia	11	Milano, Bergamo, Brescia, Como, Crema (CR), Cremona, Pavia, Lecco, Lodi, Mantova, Varese
Liguria	1	Genova
Trentino Alto A.	2	Bolzano, Trento
Veneto	8	Agordo (BL), Belluno, Rovigo, Padova, Treviso, Venezia, Vicenza, Verona
Friuli V. G.	4	Gorizia, Pordenone, Trieste, Udine
Emilia Romagna	8	Bologna, Cesena (FC), Forlì, Ferrara, Piacenza, Parma, Ravenna, Reggio E., Rimini
Toscana	8	Arezzo, Firenze, Lido Di Camaiore (LU), Ortoferale (LI), Pisa, Prato, Pistoia, Siena
Marche	5	Ancona, Fabriano (AN), Ascoli Piceno, Grottammare (AP), Fano (PU)
Umbria	1	Perugia
Lazio	2	Roma, Sora (FR)
Abruzzo	5	L'Aquila, Sulmona (AQ), Chieti, San Salvo (CH), Teramo
Campania	4	Napoli, Pignataro Maggiore (CE), Casanuovo di Napoli (NA), Pagani (SA)
Basilicata	1	Potenza
Puglia	3	Bari, Foggia, Lecce
Calabria	3	Cosenza, Gioia Tauro (RC), Reggio Calabria
Sicilia	6	Agrigento, Messina, Milazzo (ME), Palermo, Ragusa, Siracusa
Sardegna	3	Cagliari - Abbasanta-Ghilarza (OR), Sassari

È interessante confrontare la consistenza del fenomeno nazionale con dati ricavati da indagini a carattere internazionale. Come mostrano i numeri delle tabelle 1.3 e 1.4, l'Italia si colloca al quarto posto nel mondo, rivelandosi allineata con quello che risulta essere un fenomeno a carattere prettamente europeo: in Europa sono in continua crescita le conoscenze e le capacità di sviluppo di software OS, pur rimanendo gli Stati Uniti il paese con la maggior capacità di attrazione tecnologica. Bisogna peraltro notare che i progetti in lingua italiana non superano nel mondo l'1%.

Questo ultimo è comunque un dato puramente quantitativo, che non inficia la qualità del lavoro “nazionale” svolto. Sono infatti disponibili in lingua italiana tutti i migliori e più diffusi prodotti OS presenti sul mercato, oltre a prodotti gestionali e di altra natura.

**Tabella 1.3: sviluppatori software OS per nazionalità**

	Sviluppatori OS per Nazionalità	Presenza di sviluppatori OS per Paese	Capacità di attrazione
Francia	16,4 %	15,4 %	-1,4%
Germania	12,4 %	12,6 %	+0,2%
Stati Uniti	10,3 %	12,7 %	+2,4%
Italia	7,8 %	8,0 %	+0,2%
Spagna	6,7 %	6,6 %	-0,1%
Regno	6,5 %	6,5 %	-
Olanda	6,5 %	6,4 %	-0,1%
Belgio	4,0 %	3,5%	-0,5%
Svezia	3,5 %	3,5%	-



**Tabella 1.4: ripartizione di progetti OS nel mondo per lingua**

Lingue	Numero progetti	Val %
Inglese	38190	77,40%
Tedesco	3787	7,68%
Francese	2358	4,78%
Spagnolo	1541	3,12%
Russo	725	1,47%
Italiano	476	0,96%
Portoghese	418	0,85%
Giapponese	335	0,68%
Olandese	316	0,64%
Polacco	265	0,54%
Cinese	195	0,40%
Svedese	142	0,29%
Cecoslovacco	94	0,19%
Altre Lingue	500	1,01%
<b>Totale</b>	<b>49342</b>	<b>100,0%</b>

Fonte : Elaborazione ware.it su dati Source Forge.net dic. 2003

Uno studio più approfondito, relativo all'Italia, ha messo in evidenza che le forme associative, il capitale umano e di conoscenze connesse al fenomeno dell'OS vanno maturando, facilitando la creazione e la aggregazione di comunità professionali capaci di realizzare sistemi relazionali più complessi.

La crescita di queste entità avviene secondo un modello di tipo reticolare, ovvero con la nascita di poli di aggregazione a valenza distrettuale, che si evolvono nel tempo favorendo una concentrazione crescente di conoscenza e di forze, creando sinergie con ambienti universitari e imprenditoriali, seguendo una direzione che dalla periferia e dagli ambiti locali li porta alla conquista di nicchie significative anche in ambito centrale del Paese.

Tutto questo non si trova attualmente sul territorio nazionale con una distribuzione uniforme e capillare, ma si focalizza soprattutto intorno alle maggiori aree metropolitane e ai tradizionali centri di eccellenza accademica italiana.

È dal fronte dello sviluppo di software OS che, nonostante la depressione generale del mercato, sembrano venire le uniche indicazioni positive riscontrabili nelle indagini più recenti. Come auspicava il rapporto della Commissione Meo, si va delineando la presenza di nuovi soggetti, tecnologici e imprenditoriali che appaiono, almeno potenzialmente, in grado di portare nuova linfa e nuove prospettive di crescita in questo settore di mercato. Si tratta di gruppi di piccole e piccolissime imprese, di imprenditori individuali fortemente specializzati o prodotti da incubatori tecnologici o enti accademici, accomunati tutti da un unico fattore: la grande competenza sulle nuove tecnologie di sviluppo in ambiente OS.

Questa competenza, da sola, è l'elemento abilitante alla nascita di nuove filiere di produzione, in quanto non esistono altri vincoli per accedere a questo tipo di mercato. I dati presentati dal rapporto di Ware.it dimostrano già una buona presenza sparsa sul territorio nazionale, indicando nelle regioni della Lombardia, Piemonte e Lazio la maggiore concentrazione di queste PMI.

**Tabella 1:5 PMI orientate al software OS e poli di ricerca in Italia**

Regione	N°	Provincia di riferimento FS/OS	Università - Centri di ricerca e incubatori di imprese tecnologiche
Lombardia	45	Milano (29)	Politecnico Milano - Univ. Milano Dipartimento informatica e comunicazione - Acceleratore d'impresa del Politecnico di Milano - Dipartimento di Informatica e Comunicazione dell'Università degli Studi dell'Insubria
Piemonte	28	Torino (20)	Politecnico Torino - Csi Piemonte
Lazio	25	Roma (23)	Caspur Univ. Roma I - Università di Roma "Tor Vergata" - Univ. Roma I Dip. Sociologia e comunicazione
Toscana	17	Pisa (7)	Dipartimento di Informatica Univ. Di Pisa
Emilia Romagna	15	Bologna (8)	Parma: Facoltà di Matematica, AlmaCube incubatore dell'Università di Bologna
Veneto	15	Padova (6)	Dipartimento di Elettronica ed Informatica Univ. Padova VEGA, Parco scientifico tecnologico di Venezia
Liguria	8	Genova (5)	
Campania	8	Salerno ( 5)	
Marche	7	Macerata (3)	
Friuli	6	Udine (3)	Università degli Studi di Udine, Sincrotrone Trieste ScpA Area Science Park Trieste
Altre Regioni	21	Cagliari (3)	
Totale	195		Dipartimento di Informatica e Studi Aziendali Univ. Trento - Università dell'Aquila - Dipartimento di Scienze dell'Università Chieti-Pescara - Università del Dipartimento di Ingegneria dell'Innovazione, del Laboratorio di Calcolo del Dipartimento di Matematica Lecce.

Questi soggetti non si possono ancora rappresentare come una nuova industria italiana del software, ma si stanno dimostrando capaci di intraprendere iniziative significative per lo sviluppo tecnologico del Paese, sia nella Pubblica Amministrazione sia nelle PMI, quanto meno in ambito locale, favorendo la penetrazione dei sistemi e dei prodotti basati sul software OS e venendone loro stessi favoriti in un lento ma crescente ciclo virtuoso.

Questa nuova ma valida tendenza tecnologica, unita alla necessità di una più attenta valutazione del rapporto costi-benefici in fase di acquisizione di software (così come richiesto dalla Direttiva del ministro Stanca) sembra aver destato anche l'attenzione della PA centrale. Recentemente infatti sono state annunciate da molte PAC le realizzazioni o la partenza di sperimentazioni di ambienti pilota basati su software OS (qualcuna di queste verrà presentata nei prossimi capitoli).

Tutto ciò non potrà che contribuire ad amplificare gli effetti benefici che questo nuovo ecosistema sta portando, seppure in forma debole, al mercato interno della produzione del software.

A fianco, per meglio dire contrapposti, a questi nuovi soggetti protagonisti della produzione di software OS, ritroviamo nell'attuale panorama del ICT italiana la presenza dei tradizionali "big players" dell'informatica (sia nazionali che stranieri); alcuni di essi appaiono impegnati a cavalcare l'onda del nuovo mercato aperto dal software OS.

Tuttavia, nonostante i cospicui investimenti più volte pubblicamente dichiarati dai “*big players*”, i progetti e le realizzazioni importanti a loro assegnati, è attualmente difficile trovare documentazione o riferimenti sulle soluzioni rilasciate da questi attori, così come vorrebbero il paradigma e la metodologia OS. Ciò svantaggia la penetrazione e diffusione del software OS, e influisce negativamente sulle potenzialità di riuso che questa modalità di sviluppo comporta.

In questo scenario, la PA può interpretare, come in altre occasioni, un ruolo da protagonista. Non è una sorpresa scoprire che l’intero mercato ICT nazionale si aspetta che, così come è sempre avvenuto, la PA svolga la funzione di volano e contribuisca quindi al decollo e alla crescita del mercato di prodotti basato su software OS, favorendo soprattutto la nascita e la crescita di piccole e medie industrie del software nazionali.

Secondo una ricerca condotta dal ELISS, le PMI intervistate, tutte coinvolte nel mercato del software OS, ritengono che nel 2005:

- il software di tipo OS, nella PA, rappresenterà circa la metà del mercato totale server e circa un quarto del mercato totale client;
- la quota di mercato del software OS continuerà a essere maggiore sul lato server che sul lato client, com’è attualmente;
- la pubblica amministrazione giocherà un ruolo cruciale nell'affermazione dei nuovi standard.

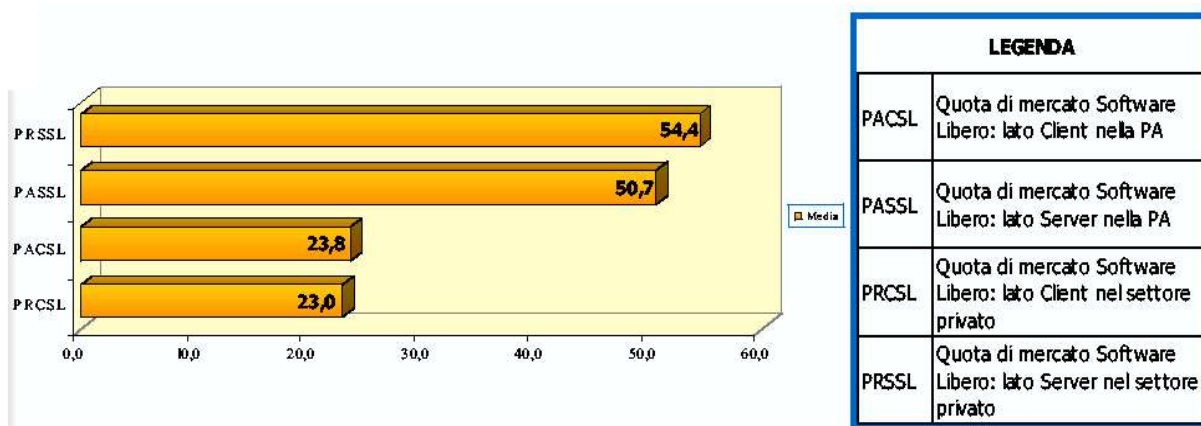


Figura 1.2: aspettative di evoluzione del mercato OS nella PA (fonte Eliss 2003)

In conclusione, i risultati delle rilevazioni citate sembrano dimostrare che i segnali che vengono dal mercato e dal mondo politico sono interpretati positivamente dagli addetti ai lavori. Costoro, pur a partire da posizioni e approcci diversi, si stanno già organizzando per poter cogliere al meglio le opportunità di business e di innovazione che si possono intravedere nello scenario che si va delineando.

## 1.2.Mercato ICT Italia

Si riportano nel seguito alcuni dati di sintesi tratti dal rapporto Assinform 2004.

Nel 2003, il mercato dell'informatica è risultato pari a 19.396 milioni di euro, in calo del 3,2% rispetto al 2002, anche per effetto di una caduta dei prezzi che ha fatto regredire non solo il comparto dei sistemi, ma anche quello del software e dei servizi (-2,2%), da sempre in crescita.

Nel 2003, il mercato dell'informatica ha accentuato le criticità del 2002, quando per la prima volta dopo anni di crescita a due cifre, si è bloccato. I 19.396 milioni di euro del 2003 indicano un ulteriore e più accentuato calo della domanda. È la conferma di una situazione difficile contrassegnata da una domanda fiacca e da una caduta dei prezzi che interessa l'intero mercato. Significativa è anche la riduzione del numero e dell'importo di nuovi progetti, sia presso l'utenza pubblica che quella privata.

Nel corso del 2003 è venuta a mancare la spinta di tutti gli attori. La domanda delle famiglie è risultata pari a 791 milioni di euro (-3,5%), quella delle piccole imprese a 3.633 milioni di euro (-4,5%), quella delle medie a 4.513 (-3,0%) e quella delle grandi imprese a 10.459 (-2,8%).

Le vendite di software e servizi, pari a 13.378 milioni di euro, sono per la prima volta regredite (-2,2%) dopo più di venti anni di crescita costante. Più in particolare, la componente servizi (sviluppo applicativo e progettuale, integrazione di sistemi, servizi di elaborazione in conto terzi, consulenza, formazione e così via ) ha fatto registrare un volume d'affari pari a 9.371 milioni di euro (in calo del 4% sull'anno prima, quando ancora risultava in crescita del +3,4%). Per contro, la componente software si è attestata a quota 4007 milioni, con una crescita del +2,2 %, vicina a quella dell'anno precedente (+3%).

La crescita del software deriva dal lancio sul mercato di nuovi strumenti d'integrazione, dall'aggiornamento di applicativi integrati (ERP) e dall'investimento in aree specifiche come la sicurezza, la business intelligence e il CRM.

Il calo dei servizi è la risultante fra la caduta delle componenti della domanda legate al lancio di nuovi progetti (servizi progettuali e implementativi per integrazione di sistemi, consulenza e formazione) e alla manutenzione applicativa, non controbilanciato dal ritrovato dinamismo della domanda di servizi in outsourcing, che è l'unica che cresce a tassi apprezzabili, sia pure moderatamente (+2,6%).

### **1.3. Il software open source: stato dell'arte**

Una delle peculiarità del software OS è che la sua diffusione è prevalentemente basata sullo scaricamento da internet e la libera copia, mentre solo in alcuni casi si ha una vendita di licenze. Per questo motivo, valutazioni di penetrazione e di quote di mercato dei prodotti non sono di facile elaborazione.

Con questa premessa, e limitatamente ad alcuni prodotti OS significativi, sono state raccolte alcune stime giudicate attendibili, tratte sia da analisi fornite da gruppi come Gartner, Forrester Research o Netcraft, sia da studi effettuati da riviste del settore e pubblicate sui siti delle stesse.

#### **1.3.1. Analisi "Hype Cycle"**

Gartner Group ha introdotto alcuni anni fa la curva denominata "Hype Cycle" applicabile alle nuove tecnologie informatiche, nella quale vi è inizialmente un picco prodotto dalle eccessive aspettative sulla tecnologia, seguito da un avallamento dovuto alla disillusione.

In seguito la curva ricomincia a salire lentamente allorché le tecnologie diventano più efficienti, meglio testate e più ampiamente accettate, tendendo a un livello quasi costante detto “plateau of productivity”.

Lo stesso Gartner Group ha condotto recentemente lo studio “Hype Cycle for Open Source Technologies, 2003” in cui ha valutato il grado di maturità di 17 tecnologie open source, ed esaminato le potenzialità di impatto delle medesime sul mercato.

Si riportano nel seguito alcune di queste valutazioni, suddivise per tipologia di soluzioni.

### ***Applicazioni per il desktop***

Definizione: suite di produttività personale comprendenti elaboratore testo, foglio di calcolo, presentazioni, client di collaborazione e scambio di messaggi.

Tempo di plateau / velocità di adozione: da due a cinque anni.

Giustificazione per la posizione sulla curva di Hype Cycle: alcune applicazioni per desktop hanno incontrato interesse da parte della stampa e dagli utenti che sono in cerca di alternative a basso costo alle offerte commerciali. Sebbene in fase di maturazione veloce, per la maggior parte degli utenti la migrazione non si presenta priva di problematiche.

Aree di impatto: la migrazione in massa verso soluzioni open source per il desktop cambierebbe lo scenario economico. Altamente improbabile a breve termine, ma con un alto potenziale di sconvolgimento.

Selezione di prodotti e fornitori: Gnome, KDE, Mozilla, Open Source Applications Foundation (OSA Foundation), Sun Microsystems, Ximian.

### ***Application Server***

Definizione: middleware open source che risiede tra il sistema operativo da un lato e risorse esterne come DBMS, comunicazione e servizi Internet dall'altro. Agiscono come contenitori della logica di business dell'utente, e forniscono servizi di sicurezza, manutenzione, accesso ai dati e altro.

Tempo di plateau / velocità di adozione: da due a cinque anni.

Giustificazione per la posizione sulla curva di Hype Cycle: posizione bene consolidata per alcune piccole web application. Cominciano a notarsi anche come alternative agli application server commerciali basati su J2EE.

Aree di impatto: accelereranno la disponibilità come commodity nel mercato degli application server e metteranno sotto pressione i prezzi per le soluzioni commerciali.  
Selezione di prodotti e fornitori: Jboss, Zope, Tomcat.

### ***Linux on the Desktop***

Definizione: uso di Linux a livello enterprise per i desktop, non solo per i server.

Tempo di plateau / velocità di adozione: da due a cinque anni.

Giustificazione per la posizione sulla curva di Hype Cycle: dopo parecchie false partenze, “Linux on the desktop” sta cominciando a guadagnare attenzione. L’aumentata stabilità delle distribuzioni di Linux può aiutare a spostare il costo totale di possesso in favore di Linux. Rimangono comunque molte questioni aperte dal punto di vista del supporto.

Aree di impatto: riduzione dei costi operativi, aumentata stabilità dei client.

Selezione di prodotti e fornitori: Lindows.com, Red Hat, SuSe, Ximian, Mandrake.

## **Database**

Definizione: database relazionali open-source.

Tempo di plateau / velocità di adozione: da due a dieci anni.

Giustificazione per la posizione sulla curva di Hype Cycle: sebbene largamente diffusi nelle installazioni medie e piccole, specialmente come parte di infrastrutture web, i database open source generalmente mancano di caratteristiche di alto livello.

Aree di impatto: l’uso di tecnologie open source per i database continuerà. Non si tratta di un fatto altamente sconvolgente di per sé, ma di un altro tassello di una più ampia infrastruttura basata sull’open source.

Selezione di prodotti e fornitori: MySQL, PostgreSQL, SAP (SAP DB).

### **1.3.2. Altre analisi**

Si riportano nel seguito i dati di alcuni specifici prodotti OS, ripresi da varie fonti.

## **Linux**

Secondo una ricerca Gartner del 2003, Windows ha consolidato la sua quota di mercato nei database, ma la crescita di Windows come piattaforma per database è tallonata dalla crescita della piattaforma Linux sia in termini puramente monetari che percentuali.

La quota di mercato di Windows nei database (principalmente basata sulle vendite di SQL Server) è salita dal 44,3% del 2002 al 47,3% del 2003. Linux, d’altra parte, è cresciuto a un ritmo rovente come piattaforma per database: 153%. (*fonte: Scott Bekker, ENTNews, 27/05/04*)

Secondo un recente rapporto della Forrester Research Inc. (aprile 2004), in cui sono state intervistate 140 grosse compagnie nordamericane, oltre il 60% dei rispondenti hanno affermato di utilizzare attualmente prodotti open source o di stare pianificando di farlo. In particolare, il 53% utilizzano Linux per applicazioni mission critical, il 52% scelgono Linux per le nuove applicazioni, il 44% stanno utilizzando Linux per il porting di vecchie applicazioni su nuovo hardware, e il 33% utilizzano applicazioni che richiedono Linux.

## **Apache**

Dati Netcraft del maggio 2004 confermano la dominanza di Apache nell'ambito dei web server, con una percentuale del 67% su un totale di 50.550.965 siti censiti, mentre il web server Microsoft si attesta sul 21% e SunONE sul 3,25%.

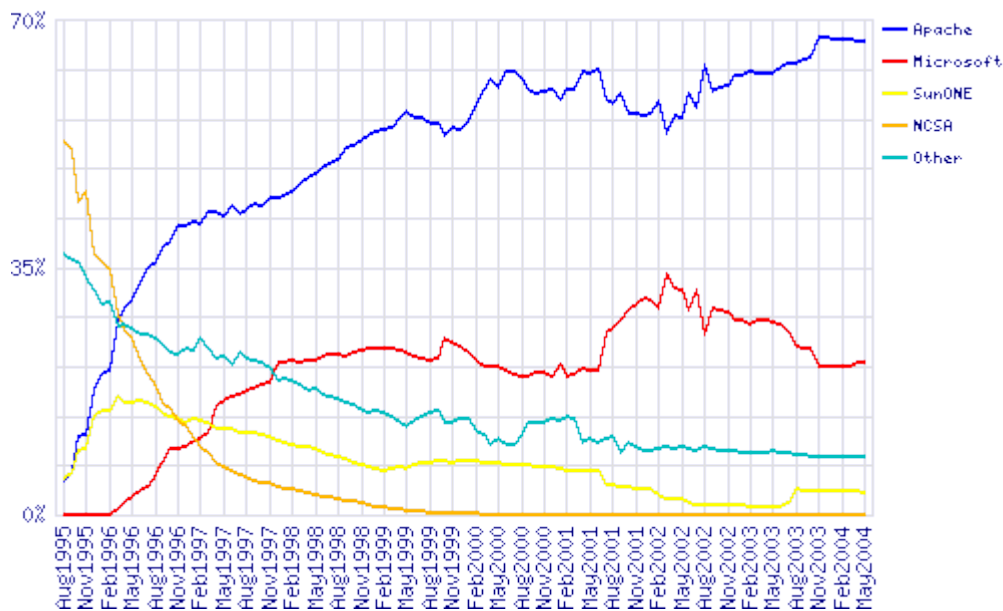


Figura 1.3: ripartizione dei web server - agosto 1995/maggio 2004

## MySQL

Numero di download: 20 milioni

Costo medio annuale per licenze e supporto: \$4,482

Siti MySQL totali con licenza : 4,000 (approssimativamente), con 4 milioni di utenti attivi

*Fonte: Computerworld, MySQL AB, Meta Group Inc. Ottobre 2003*

Secondo la citata indagine di Forrester Research, il 52% degli intervistati sta utilizzando o pensa di utilizzare il database MySQL.

L'articolo "Open Source Databases Gain Ground" di Joe McKendrick del maggio 2004, pubblicato sul sito "Database, trends and applications" (DBTA), riporta una indagine recente di Evans Data, secondo la quale SQL Server e Access continuano a dominare, ma i database open source tendono ad aumentare la propria presenza. Sia SQL Server che Access sono cresciuti di circa il 6% nell'ultimo anno, mentre l'uso di MySQL è aumentato di più del 30%.

## Jboss

Entro il 2008 si prevede che, per numero di licenze installate su server di produzione, l'open source J2EE (Jboss) sarà inferiore soltanto a due o tre prodotti commerciali (probabilità 0.7). Entro il 2008, più del 30% di installazioni di piattaforme J2EE certificate saranno con licenza OS, oppure inserite in prodotti più grandi e di costo più elevato (probabilità 0.8). *Fonte: Gartner, maggio 2003*

## **OpenOffice.org / StarOffice**

Negli Stati Uniti, secondo uno studio Forrester Research dell'aprile 2004, su 140 grosse imprese del Nord America, 12 di esse stanno già utilizzando OpenOffice.org, mentre 8 stanno pianificando il suo prossimo utilizzo. Ciò significa una quota di mercato per OpenOffice.org pari attualmente al 8,6%, mentre, poiché una migrazione di suite di produttività può richiedere da 3 a 6 mesi, realisticamente le rimanenti imprese effettueranno il passaggio entro la fine del 2004. Considerando quindi che si avranno 20 aziende su 140 utilizzanti OpenOffice.org entro fine 2004, si può prevedere una quota di mercato del 14,3% per quanto riguarda il settore delle grosse imprese americane.

Si riportano inoltre alcuni dati tratti dal report "Comparative assessment of Open Documents Formats" dell'IDA, dicembre 2003. Il Ministero Francese degli Interni ha già migrato a OpenOffice.org 15.000 desktop, e intende migrarne altri 15.000, sui 100.000 che possiede, entro un anno a decorrere dal novembre 2003. Regno Unito, dicembre 2003: il governo britannico ha stipulato un contratto per installare StarOffice su 500.000 computer. Germania: la città di Monaco di Baviera, che è la terza città tedesca in ordine di grandezza, ha scelto OpenOffice.org per più di 15.000 desktop.

Altri paesi e amministrazioni che stanno adottando OpenOffice.org includono la Cina, Thailandia, Israele, Australia, Filippine, Uganda e Vietnam. I principali criteri in favore dell'adozione sono il prezzo (nullo o estremamente basso), l'essere open e la disponibilità multiplatforma.

Da tenere presente che il formato nativo dei dati di OpenOffice.org / StarOffice è XML-based pienamente documentato e liberamente disponibile. Inoltre tale formato sta per essere standardizzato dall'OASIS. Lo scopo dell'OASIS è di creare un formato di file aperto XML-based per le applicazioni office. A tale scopo OASIS intende utilizzare il formato di OpenOffice.org come base su cui applicare alcuni cambiamenti minori.

### **1.3.3. Uso del software OS nella pubblica amministrazione**

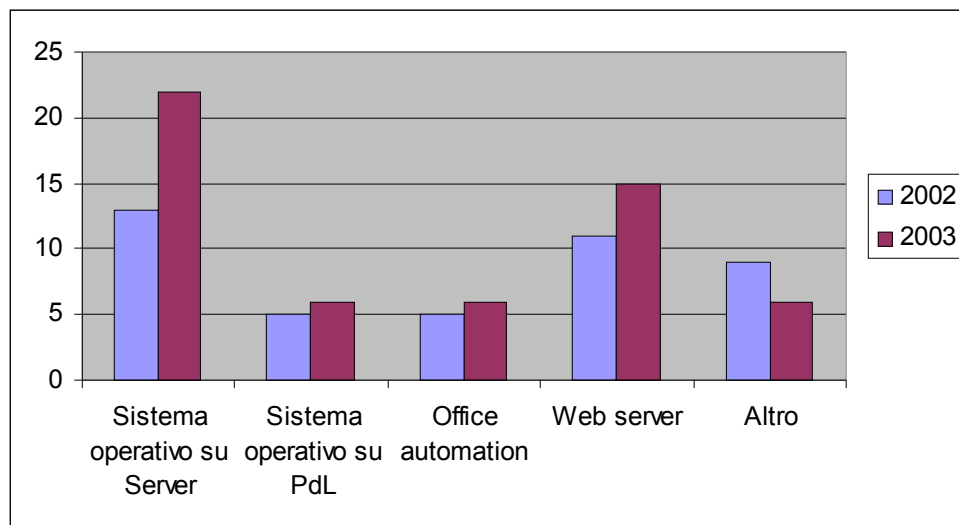
L'utilizzo di software open source nella pubblica amministrazione centrale appare in espansione. Dai dati dell'ultimo consuntivo CNIPA, il 54% delle amministrazioni (ventotto su cinquantadue, di cui 18 amministrazioni centrali e 10 Enti pubblici) utilizzano anche software open source per i propri sistemi, contro il 42% dei rispondenti dell'ultima rilevazione. In tabella 1.5 viene riportata una sintesi dei dati del consuntivo CNIPA.

**Tabella 1.6: Soluzioni open source per tipologia, anni 2002-2003**

	Amministrazioni Centrali		Enti		Tutte	
	2002	2003	2002	2003	2002	2003
Sistema operativo su Server	8	13	5	9	13	22
Sistema operativo su PdL	3	4	2	2	5	6
Office automation	4	5	1	1	5	6
Web server	9	11	2	4	11	15
Altro	6	5	3	1	9	6



In figura 1 è riportata, per le diverse tipologie di software open source, il numero complessivo di amministrazioni ed Enti che ne fanno uso nel 2003. Vengono riportati, come riferimento, gli stessi dati relativi all'anno precedente (2002).



**Figura 1.4: penetrazione del software OS nella PAC (2 anni a confronto)**

Si noti, in controtendenza, la voce “altro”, che risulta in diminuzione nell'ultimo anno rispetto al precedente, sintomo che i software open source stentano ancora ad affermarsi in ambito diverso dagli impieghi tradizionali, quali web server e sistemi operativi.

Tra le amministrazioni e gli Enti ove esistono esempi significativi di impiego di software open source nell'ultimo anno, si citano ad esempio i seguenti casi:

- La Presidenza del Consiglio dei Ministri, ove è stato sviluppato con tecnologie open source il Portale dei Servizi e, sempre con tecnologie open source, è stato realizzato un sistema di misura del traffico dati.
- Il Dipartimento Politiche Fiscali, ove è proseguita la realizzazione di strumenti di collaborazione (calendar e forum) in ambiente open source.
- L'Agenzia del Territorio, ove è stata adottata una soluzione che prevede l'utilizzo, per le applicazioni del sistema informativo catastale, di una architettura web based costituita da Linux, Apache e linguaggio PHP. Tale soluzione è attualmente in fase di estensione in tutti gli uffici.
- L'Agenzia del Demanio, ove viene utilizzato software open source per la realizzazione del servizio a supporto della gestione dei beni confiscati.
- La Scuola Superiore della Pubblica Amministrazione, ove è stata realizzata la procedura di gestione amministrativa e contabile dei corsi (GACC) in ambiente Linux con database MySQL. Pur proseguendo l'uso della piattaforma applicativa Microsoft, la Scuola ha consolidato un ambiente parallelo costituito da piattaforme di tipo open source e soluzioni applicative ed architetturali che privilegiano le strutture informative web based. Il sito Internet è realizzato in PHP, usa il database MySQL, ed è ospitato su server Linux. Anche il nuovo sito Intranet usa la medesima piattaforma. La reingegnerizzazione delle procedure gestionali per la

realizzazione del portale interno delle procedure, ha adottato il medesimo ambiente e analoghe modalità di sviluppo (piattaforma PHPNUKE).

- L' INPS sta utilizzando un motore di ricerca a reti neurali, sviluppato con tecnologia open source, in alcuni progetti di Knowledge Management, tra cui ad esempio il progetto "NEUWEB" (Neural Network Engineered Web Portal), implementazione di un portale per l'accesso a informazioni di sicurezza sociale.
- La SOGEI utilizza, per il sistema di accesso tramite GPRS alla rete aziendale, un server RADIUS open source, un DNS basato su Linux, OpenSSL su macchine Linux per l'emissione di certificati digitali e per la cifratura dei dati, un server OpenLDAP per l'autenticazione degli accessi al web server Apache, un proxy server SQUID, un news server INN.

Per quanto riguarda le amministrazioni locali, si citano i dati forniti da Rur, Formez, Censis riguardanti l'impiego di sistemi operativi e web server open source (vedi tabella 1.1).

Nel numero 74 della propria newsletter istituzionale, in data 5 aprile 2004, il Ministro per l'Innovazione e le Tecnologie ha diffuso una richiesta di segnalazioni sul tema dell'open source, invitando i lettori a segnalare progetti e azioni promossi dagli uffici della pubblica amministrazione tesi a favorire l'uso di software a codice sorgente aperto. Da tale data, il modulo per la segnalazione è rimasto disponibile sul sito del MIT, all'indirizzo [http://www.innovazione.gov.it/ita/newsletter/segnalare\\_os.shtml](http://www.innovazione.gov.it/ita/newsletter/segnalare_os.shtml).

A distanza di circa due mesi, hanno risposto ed inviato segnalazioni 21 pubbliche amministrazioni, di cui 20 locali e una centrale. Diciotto amministrazioni hanno descritto uno o più progetti in corso (o in fase di avvio) che prevedono l'uso di prodotti open source. Essi riguardano soprattutto la creazione di siti web e di infrastrutture di rete, ma anche la realizzazione o la re-ingegnerizzazione di applicazioni gestionali e di produttività individuale. Tre amministrazioni hanno segnalato semplicemente un loro interesse in materia open source, oppure hanno segnalato iniziative di tipo conoscitivo, comunicativo o didattico riguardante lo OS. Hanno inoltre inviato una segnalazione 17 soggetti privati, tra cui consulenti, aziende e consorzi che offrono prodotti o servizi in ambito OS. Tra questi, anche aziende che già in passato hanno collaborato con il CNIPA (ad esempio iCube).

In tutto, dunque, i rispondenti sono stati 38. Si ritiene che tale numero sia degno di nota, dato il breve tempo trascorso e la scarsa conoscenza dell'iniziativa. Una piccola minoranza di segnalazioni (dell'ordine del 10% del totale) è accompagnata da richieste di supporto di varia natura, più di genere contrattualistico/giuridico che tecnologico.

### **1.3.4.Evoluzione tecnica e benchmark**

#### *Linux*

Il kernel è giunto alla versione 2.6.x, che presenta notevoli miglioramenti sia in termini di prestazioni che di scalabilità rispetto alle precedenti versioni.

Nell'impiego come server, è da notare la possibilità di gestire fino a 64 processori. Il numero massimo di utenti passa da 65 mila a oltre 4 miliardi, con 1 miliardo di processi concorrenti su uno stesso sistema.

Il supporto nativo della Native POSIX Thread Library (NPTL) permette l'esecuzione, in modo efficiente e stabile, di applicazioni che utilizzano processi molteplici contemporaneamente (multithreading), consentendo di ottimizzare l'impiego di sistemi multiprocessore.

Il server beneficia, inoltre, dell'introduzione di un sistema di ACL (Access Control List) per i filesystem EXT2 ed EXT3 e di una maggiore interoperabilità con Windows grazie ai miglioramenti apportati ai supporti del Logical Disk Manager e al file system NTFS in lettura e in scrittura.

Red Hat ha ottenuto la certificazione “*Common Operating Environment*” (COE)<sup>3</sup> dal Governo USA<sup>4</sup> per il sistema operativo Red Hat Linux Advanced Server.

Sia IBM<sup>5</sup>, che Oracle<sup>6</sup> si sono impegnate a collaborare con la comunità OS affinché Linux possa ottenere la certificazione Common Criteria<sup>7</sup> (CC)<sup>8</sup>, certificazione di sicurezza riconosciuta dai più importanti Paesi occidentali, fra cui l'Italia, e richiesta da molte Agenzie Governative e dai Militari come prerequisito per l'adozione di infrastrutture informatiche.

### *MySQL*

Il settimanale *Week.it*, in data 20/3/2002, riportava i risultati di un confronto tra cinque importanti RDBMS:

- Ibm Db2 7.2 con Fix Pack 5,
- Microsoft Sql Server 2000 Enterprise Edition con Service Pack 2,
- MySQL 4.0.1 Max,
- Oracle 9i Enterprise Edition 9.0.1.1.1,
- Sybase Adaptive Server Enterprise (ASE) 12.5.0.1.

L'obiettivo del confronto era verificare il comportamento dei suddetti RDBMS, usati insieme con un application server Java.

---

<sup>3</sup> Per maggiori informazioni sulla certificazione COE: <http://diicoe.disa.mil/coe/>

<sup>4</sup> Red Hat's Advanced Server version received on February 13, 2003 the Defense Department's certification running on an Intel-based IBM server, the first version of Linux to pass the milestone. Red Hat is trying to coax customers to move as quickly as possible from its less-expensive products to the better-supported Advanced Server version.

<sup>5</sup> IBM conta, in un arco di tempo di 10 mesi e con una spesa di 1 milione di dollari, di ottenere la certificazione CC per la distribuzione di Red Hat che gira sui sistemi eServer. La IBM conta di estendere i suoi sforzi per certificare altre piattaforme basate su Linux, quindi si è impegnata a sviluppare nuove funzionalità di sicurezza da aggiungere al *kernel* di Linux e lavorare a stretto contatto con i distributori di Linux per agevolare la certificazione dei loro prodotti.

<sup>6</sup> Red Hat and Oracle have teamed to get Linux evaluated under the Common Criteria, a certification that could open doors for the broader use of open-source software by government agencies. The effort is expected to take nine to 10 months and cost up to \$1 million. But if successful, it could pay off handsomely for Red Hat and Oracle, as well as for Linux.

<sup>7</sup> Il CC è un criterio attraverso cui viene valutato - a livello implementativo - il grado di sicurezza e di affidabilità di un prodotto tecnologico. Il CC è ad esempio requisito fondamentale affinché un prodotto informatico possa essere usato negli Uffici Governativi Americani che svolgono una qualche attività nell'ambito della sicurezza nazionale.

<sup>8</sup> Per maggiori informazioni sui Common Criteria: <http://csrc.nist.gov/cc/> e <http://www.commoncriteria.org/>

Nei test si sono potuti valutare anche i diversi tipi di tuning che permettono di migliorare le prestazioni di ognuno di questi prodotti.

Nel complesso, le migliori prestazioni e la migliore scalabilità sono state offerte da Oracle9i e da MySQL, anche se il prodotto di Oracle è rimasto quasi sempre in leggero vantaggio.

Fino a un carico di circa 200 utenti Web, tutti e cinque i prodotti si sono comportati in modo abbastanza simile. A questo punto, però, i notevoli problemi del suo driver Jdbc (Java database connectivity) hanno costretto MS SqlServer a fermarsi a un throughput di circa 200 pagine al secondo, mentre il throughput degli altri sistemi continuava a crescere.

Una volta raggiunto un carico di circa 500 utenti Web, le prestazioni di ASE si sono assestate a circa 500 pagine al secondo, mentre quelle di Db2 sono precipitate fino a stabilizzarsi intorno alle 200 pagine al secondo.

Le prestazioni di Oracle9i e di MySQL, invece, hanno continuato a crescere fino ad assestarsi a circa 600 pagine al secondo.

In generale, le prove hanno dimostrato che un tuning manuale dei singoli database permette di ottenere prestazioni notevolmente migliori, fino a raddoppiare il throughput offerto dalla configurazione predefinita.

Gli elementi sui quali gli utenti possono intervenire sono essenzialmente tre: la gestione della memoria da parte del database, i driver utilizzati dalle diverse applicazioni per l'accesso ai dati e l'organizzazione dei dati all'interno del database.

Ottimizzare la gestione della memoria di un database server non è semplice, perché si deve assegnare la giusta quantità di memoria a ogni componente del sistema.

Da questo punto di vista, i prodotti più semplici da configurare sono stati Sql Server e MySQL, mentre Oracle9i è stato sicuramente il più complesso, per via delle diverse cache che si possono gestire.

Sql Server, MySQL e ASE hanno richiesto circa 50Kbyte di Ram per ogni connessione utente, mentre Db2 ne ha richiesti circa 177Kbyte.

Oracle9i, invece, ha richiesto circa 400Kbyte di Ram per ogni connessione utente, lasciando così una quantità molto minore di memoria per le diverse cache per i dati e per i piani di esecuzione delle interrogazioni e delle stored procedure.

Le ottime prestazioni di MySQL sono dovute in gran parte anche a una delle novità della versione 4.0.1: una cache estremamente veloce che conserva i risultati delle query già eseguite.

La versione 4 di MySQL supporta le transazioni, la 4.1 le subqueries, ed è stata inoltre sviluppata una tecnologia per il clustering denominata MySQL Cluster. La versione 5.0, prossima all'uscita, supporta le stored procedures basate sul framework SQL-99, che è simile al PL/SQL di Oracle, oltre a un elementare supporto per i cursori.

### *PostgreSql*

Si riporta un estratto di un articolo del settimanale *Week.it* del 27/3/02:

*“La versione 7.2 del famoso database si rivela un prodotto di classe decisamente enterprise: molto solido e adatto alla gestione di grossi volumi di dati.”*

PostgreSQL è un progetto open source, nato nel 1996 con l'intento di creare un database enterprise-oriented, robusto e flessibile, sfruttando i noti vantaggi della pubblicità del codice sorgente, della gratuità dello stesso e di una grossa base di sviluppo.

Con la recente release 7.2, è interessante metterlo a confronto con i suoi omologhi open source e con molti prodotti commerciali.

PostgreSQL è un database realizzato secondo le specifiche Sql 92 e, quindi, fortemente orientato alle transazioni. Ha un supporto all'indicizzazione sotto forma di trigger e un preprocessore Sql nativo, oltre che funzioni complete per la gestione di join tra tabelle.

Dal sito di riferimento ([www.it.postgresql.org](http://www.it.postgresql.org)) è possibile scaricare il codice da compilare e i binari precompilati.

Le piattaforme supportate sono molteplici, spaziando dagli Unix commerciali quali Solaris, Hp-Ux, Aix e Irix, fino ai cloni di tipo free quali Linux e FreeBSD.

.....  
Con la versione 7.2, PostgreSQL si conferma un database molto robusto e bene implementato, adatto alla gestione di grossi volumi di transazioni. La completa interfacciabilità e una migliorata logica di pulizia della base dati lo rendono decisamente consigliabile anche in contesti enterprise.

### *Samba*

Samba è un file e print server open source, che consente di fornire tali servizi a PC basati su Windows.

Secondo test condotti da IT Week Labs nell'ottobre 2003, l'ultima versione (Samba 3) risulta 2,5 volte più veloce di Windows Server 2003, nello stesso ruolo. La differenza di performance è cresciuta enormemente rispetto al confronto tra Samba 2 e Windows Server 2000, come pure la scalabilità.

Il test è stato condotto utilizzando un server HP ProLiant BL10 e il Ziff-Davis NetBench file server benchmark con 48 client system, senza utilizzare strumenti di tuning per il software.

### **1.3.5.Sintesi dello scenario del software OS**

Le soluzioni open source stanno avendo una penetrazione sempre maggiore sia nel settore pubblico che nel privato (in questo secondo caso probabilmente più all'estero che in Italia). In particolare i prodotti che già erano abbastanza affermati (Apache, MySQL, Linux) stanno subendo un consolidamento tecnologico e di supporto che li porta sempre di più ad essere considerati eleggibili per un utilizzo professionale e mission critical. Parimenti cresce a livello enterprise l'interesse per soluzioni desktop open source alternative a Windows (il cosiddetto *Linux on the desktop*).

Il livello di diffusione tuttavia sembra ancora insoddisfacente, e la distribuzione dei progetti disomogenea sul territorio: ciò sembra dovuto a numerosi fattori, tra cui il più importante probabilmente è la mancanza di conoscenze adeguate delle tipologie di software in argomento.

## **1.4. Iniziative pubbliche UE ed extra UE**

### **1.4.1. Istituzioni UE**

Nel 2002 la Commissione Europea ha prodotto il report “[Pooling Open Source Software](#)” che tratta dei risparmi ottenibili condividendo il software su basi open source. Lo studio raccomanda di creare un sistema ove il software sviluppato per il settore pubblico possa essere raccolto su base di donazione per un successivo riutilizzo.

Nell'agosto 2003 è stata attivata l'iniziativa denominata “Encouraging good practice in the use of open source software in the public administrations”. Tale iniziativa comporta principalmente lo sviluppo di centri di competenza nazionali e regionali per facilitare lo scambio di informazioni sulle opportunità e i rischi connessi con l'open source.

Nel documento “Comunicazione della Commissione al Consiglio, al Parlamento Europeo, al Comitato Economico e Sociale al Comitato delle Regioni” del settembre 2003 viene riportato nell'elenco delle azioni da perseguire: “Promuovere lo scambio di esperienze sull'uso di norme aperte e open source tra le amministrazioni, tra l'altro mediante i pertinenti programmi dell'UE”

Il sesto programma quadro recita testualmente:

#### *Implementation and tasks:*

*Based on existing activities or using examples from other Member States, national inventories of projects or of selected content will be developed. The inventories need to comply with quality standards for the content to be included (based on good practice guidelines - see action 3b) and for the access services they provide. A technical workgroup led by Member States with greatest experience in this area will define the possible platforms paying particular attention to open-source and free software and produce a set of guidelines to support other Member States in establishing usable and sustainable inventories.*

La promozione all'utilizzo del software OS nel settore pubblico e la diffusione di best practices nell'eGovernment avviene principalmente attraverso i programmi IDA e IST (Information Society Technologies). In quest'ultimo sono implementati gli aspetti di ricerca e di sviluppo relativamente al software OS. Attualmente vi sono circa una ventina di progetti in corso di sviluppo. Sul sito web della Commissione Europea, nella sezione open source, si trova un elenco dei progetti OS finanziati dalla Commissione, ed è possibile scaricarne il catalogo.

Il programma IDA (Interchange of Data between Administrations) della Commissione Europea dispone sul suo sito web di una sezione “[Open source observatory](#)” (OSO) di cui si danno più dettagliate notizie nel capitolo dedicato ai centri di competenza.

Meritevole di particolare nota è il documento pubblicato dall'IDA stesso “[IDA Open Source Migration Guidelines](#)”, cui si accompagna un “comparison spreadsheet” (in

formato xls), il cui intento è permettere alle amministrazioni di calcolare i costi di migrazione verso una istallazione completamente OS a partire da una proprietaria: non è un modello per calcolare il TCO (total cost of ownership), ma compara i costi di una istallazione proprietaria con quello di una OS, insieme ai costi di migrazione, prendendo in considerazione particolare conto i costi significativamente differenti nei due casi.

Il [Joint Research Centre](#) della Commissione Europea gestisce un certo numero di progetti open source:

- [GIST](#), un generico information server per ambienti di collaborazione;
- [PERLZ](#), una implementazione Perl del protocollo Z39.50;
- [RDFStore](#), un parser RDF e database in Perl;
- [Open-GIS Web Mapping](#), una implementazione Java del protocollo OpenGIS web mapping.

È stato costituito recentemente il consorzio COSPA<sup>9</sup> (al quale il CNIPA ha aderito in qualità di osservatore), con lo scopo di analizzare gli effetti dell'introduzione degli standard aperti per il formato dei dati (Open Data Standard: ODS) e del software open source per la produttività personale e la gestione dei documenti nelle pubbliche amministrazioni europee. È possibile prendere parte ai lavori del consorzio con attività sperimentali. Il Consorzio è finanziato nell'ambito del sesto programma quadro della UE per un importo pari a 2,6 milioni di euro (durata del progetto da gennaio 2004 a dicembre 2005). Attualmente raccoglie 15 partecipanti da parte di Italia, Ungheria, Danimarca, Regno Unito, Irlanda, Belgio. Inoltre partecipano tre osservatori internazionali: Canada, Nuova Zelanda, UNESCO.

### **1.4.2. Paesi UE**

#### **Danimarca**

Il 20 giugno 2003 il Ministro Danese per la Scienza, la Tecnologia e l'Innovazione ha pubblicato il documento "The Danish Software Strategy"<sup>10</sup>. In base ad esso è stato stabilito che le soluzioni per il software pubblico dovranno basarsi sui principi del massimo "value for money" senza riguardo alla provenienza del software, sulla competizione, indipendenza e libertà di scelta, interoperabilità e flessibilità, sviluppo e innovazione.

Il summenzionato "strategy paper" è il seguito di una estesa analisi sul software OS intrapresa dal Danish Board of Technology e pubblicata nell'ottobre 2002. Tale analisi enfatizza la competizione addizionale portata sul mercato dal software OS, la libertà di scelta, l'interoperabilità e il maggior "value for money".

#### **Finlandia**

Il ministero finlandese delle Finanze, nell'ottobre 2003, ha rilasciato il documento:

---

<sup>9</sup> <http://www.cospa-project.org>

<sup>10</sup> <http://www.tekno.dk/subpage.php3?article=969&survey=14&language=uk&front=1>

“Recommendation on the openness of the code and interfaces of State information systems”<sup>11</sup> sull'utilizzo di software open source per lo sviluppo di applicazioni per l'amministrazione statale.

Le conclusioni cui giunge lo studio, che include il report “Procurement of tailored software for State administration - the effects of open source”, sono che l'open source è una valida alternativa, soprattutto in ragione di principi di trasparenza, e inoltre si conferma il principio che la pubblica amministrazione debba avere la proprietà del codice delle applicazioni sviluppate ad hoc e il diritto di modificarle.

È stato anche prodotto un report<sup>12</sup> sulla possibilità di utilizzo della suite OpenOffice.org e del sistema operativo Linux come ambiente standard per le workstation della città di Turku. In conclusione, il report afferma che nel breve termine è raccomandato il passaggio a OpenOffice.org entro il 2003, mentre a lungo termine l'obiettivo è di passare a Linux.

## **Francia**

L'ADAE (Agence pour le développement de l'administration électronique) è un servizio interministeriale creato il 21 febbraio 2003 e collocato presso l'ufficio del Primo Ministro. Ha preso il posto dell' ATICA, che a suo tempo aveva pubblicato il documento: “Guide de choix et d'usage des licences de logiciels libres pour les administrations”<sup>13</sup>.

Tale documento ha lo scopo di proporre alle amministrazioni un metodo di scelta e di uso delle licenze di software libero. La guida affronta la scelta e uso delle licenze di software libero in relazione alle due situazioni nelle quali le amministrazioni possono trovarsi:

- acquisto ( a titolo oneroso o gratuito) di software o di componenti;
- sviluppo di software sviluppato da o per conto delle amministrazioni.

Nel corso del programma di amministrazione elettronica 2004/2007 denominato ADELE sono previsti tre progetti attinenti a tematiche open source:

- ADELE 127: messa in opera di software libero e strumenti di sviluppo collaborativo;
- ADELE 128: piattaforma tecnica per lo sviluppo collaborativo: centri di risorse tecniche;
- ADELE 130: migrazione dei posti di lavoro.

L'ADAE mantiene inoltre un repertorio per il riuso del software e dei progetti sviluppati da varie amministrazioni dello stato francese in open source (Réutilisation des logiciels et Bouquet du libre<sup>14</sup>).

Nel 1999 è stato creato il consorzio “ObjectWeb” da parte di [Bull](#), [France Telecom R&D](#) e [INRIA](#) (l'Istituto Nazionale Francese per la Ricerca Informatica). Si tratta di una comunità open source il cui scopo è lo sviluppo di componenti middleware flessibili e

<sup>11</sup> <http://www.vm.fi/tiedostot/pdf/en/65051.pdf>

<sup>12</sup> <http://www.turku.fi/tieto/liite44.rtf>

<sup>13</sup> [http://www.adae.pm.gouv.fr/article.php3?id\\_article=172](http://www.adae.pm.gouv.fr/article.php3?id_article=172)

<sup>14</sup> <http://www.adae.gouv.fr/index.php3>



adattabili distribuiti in modalità open source. Tali componenti spaziano da infrastrutture software e protocolli specifici fino a piattaforme integrate. Nel 2002 ObjectWeb si è evoluto in un consorzio internazionale di tipo no profit aperto alle imprese, alle istituzioni e ai singoli individui<sup>15</sup>.

## **Germania**

Il KBSt è una agenzia di coordinazione e consulenza del Governo Federale Tedesco per quanto concerne l'information technology. Pubblica lettere informative e mantiene un sito web<sup>16</sup>.

Una delle sue iniziative è stata la lettera n° 2/2000 “[Open Source Software in the Federal Administration](#)”, già citata nel rapporto della commissione d'indagine. Inoltre, nel luglio 2003 è stata pubblicata una “Migration Guide<sup>17</sup>” e, recentemente, è stato costituito presso il KBSt un centro di competenza (OSS Kompetenzzentrum), di cui si danno più dettagliate notizie nel capitolo dedicato ai centri di competenza.

Un altro documento degno di nota pubblicato dal KBSt è “[Linux: An Opportunity for More Software Diversity in Public Administration](#)”.

Il Parlamento Federale Tedesco (Bundestag) sta esaminando la possibilità di utilizzare Linux su alcune sue infrastrutture informatiche e in particolare sui server<sup>18</sup>.

L'Ufficio Federale delle Finanze (Bundesamt fuer Finanzen<sup>19</sup>), dopo un periodo di valutazione di 18 mesi, ha installato Linux su due mainframes IBM: un IBM eServer zSeries z990 e un IBM eServer zSeries z900 (su quest'ultimo è installato sia Linux che z/OS) che svolge funzioni di sistema di backup.

Nel maggio 2003, il consiglio municipale della città di Monaco di Baviera ha deciso di effettuare la migrazione a Linux di circa 14.000 sistemi desktop con il supporto di IBM-SuSE. Il raggruppamento IBM-SuSE si è aggiudicato il contratto nonostante che Microsoft avesse fatto alla fine una offerta più bassa. Sembra che l'offerta di Microsoft (circa 27,3 milioni di euro) sia stata circa 3 milioni di euro più bassa di quella della controparte (da notare tra l'altro che Microsoft ha creato un fondo speciale per vendere Windows sottocosto), ma la città ha scelto ugualmente Linux in base a motivazioni strategiche. La decisione è stata motivata dal sindaco della città tedesca con la seguente considerazione: “tale decisione strategica rende Monaco meno dipendente da un singolo fornitore e pone le basi per una maggiore competizione (sul mercato)”.

Il Governo Tedesco ha inoltre deciso d'installare Linux in altri dipartimenti. In particolare, la polizia del Laender della Bassa Sassonia ha previsto di migrare a Linux 11.000 PC nel corso del 2004.

Il Ministero dell' Interno Tedesco<sup>20</sup> ha annunciato un accordo con IBM per l'acquisto di soluzioni hardware e software che supportino Linux.

## **Olanda**

<sup>15</sup> <http://consortium.objectweb.org/consortium.html>

<sup>16</sup> <http://www.kbst.bund.de>

<sup>17</sup> [http://www.kbst.bund.de/Anlage303807/pdf\\_datei.pdf](http://www.kbst.bund.de/Anlage303807/pdf_datei.pdf)

<sup>18</sup> <http://www.bundestag.de/presse/bp/2002/bp0202/0202100c.html>

<sup>19</sup> <http://www.bff-online.de>

<sup>20</sup> <http://www.bmi.bund.de/>

Nel giugno 2002 l'[International Institute of Infonomics](http://www.infonomics.nl) dell'Università di Maastricht, in partnership con Berlecon Research, ha pubblicato lo studio "FLOSS - Free/Libre Open Source Software: Survey and Study<sup>21</sup>". Tra le conclusioni dello studio, si pone in evidenza che parecchi grossi fornitori di software stanno attivamente contribuendo a progetti open source, e che il supporto pubblico all'open source dovrebbe principalmente basarsi su tre principi:

- dovrebbe essere strutturato in modo tale da lasciare decidere al mercato quali progetti sono da considerarsi utili;
- dovrebbe indirizzarsi verso quei progetti che realizzano software maggiormente utilizzabili come componenti di base;
- il tipo di licenza sotto la quale viene rilasciato il software dovrebbe essere tale da permetterne l'utilizzo nel maggior numero di casi possibile.

Varie attività sono state intraprese nel campo OS, in particolare è stato studiato il suo utilizzo al fine di ridurre la dipendenza da fornitori software esterni, di migliorare la qualità dei sistemi informativi governativi, di ridurre i costi, e di migliorare l'interscambio dati tra amministrazioni.

L'Institute for Information and Communication Technology (ICTU) ha intrapreso un programma per l' "Open Standards and Open Source Software (OSOSS)" in ambito governativo, di cui si danno notizie più dettagliate nel capitolo dedicato ai centri di competenza.

## **Norvegia**

Skolelinux<sup>22</sup> è una organizzazione norvegese che ha sviluppato una distribuzione Linux per le scuole, utilizzata anche in altri Paesi come Spagna e Germania.

## **Regno Unito**

Il 15 luglio 2002 è stato pubblicato il documento "Open source software – use within UK government<sup>23</sup>" da parte dell'Ente "Office of Government Commerce". In esso si afferma che il governo del Regno Unito:

- considererà le soluzioni OS su basi di pari dignità col software proprietario nelle forniture, e i contratti saranno basati sul principio "value for money";
- userà solo prodotti per l'interoperabilità che supportano open standard e specifiche in tutti gli sviluppi futuri per l'IT;
- cercherà di evitare di rimanere legato a prodotti e servizi proprietari;
- cercherà di ottenere pieni diritti sul codice del software sviluppato su ordinazione o sugli adattamenti del software COTS (Commercial Off The Shelf) ogni qualvolta ciò significhi ottenere il migliore "value for money";

---

<sup>21</sup> <http://www.infonomics.nl/FLOSS/report/>

<sup>22</sup> <http://www.skolelinux.no/index.php.en>

<sup>23</sup> <http://www.ogc.gov.uk/oss/OSS-policy.html>

- esplorerà ulteriormente le possibilità di utilizzare il modello OS come regola per il software di Ricerca e Sviluppo finanziato con fondi governativi.

Inoltre, nell'ottobre 2003, l'Office of Government Commerce ha annunciato piani per testare software open source in nove differenti aree dell'amministrazione. Il progetto è guidato dall'Office of the e-Envoy e misurerà i benefici dell'uso di prodotti OS rispetto a quelli proprietari.

### **Spagna**

In Extremadura<sup>24</sup>, regione del sud-ovest della Spagna, nell'aprile 2002 è stato varato un progetto basato sui principi della connettività e dell'alfabetizzazione informatica per ogni cittadino e per la migrazione di tutti i computer della regione da Windows a Linux. È stata sviluppata una versione ad hoc di Linux, chiamata GNU/Linex, per l'installazione nelle scuole e per la diffusione nella pubblica amministrazione e nelle piccole e medie imprese.

Il risparmio di 30 milioni di euro in costi di licenza sarà utilizzato per la formazione e per sviluppare l'informatizzazione dell'industria locale. Il governo locale ha inoltre investito in un centro di competenza e di sviluppo, al fine di realizzare sistemi per la finanza e la salute pubblica che saranno distribuiti gratuitamente.

### **Svezia**

L'Agenzia Svedese per la Gestione Pubblica<sup>25</sup> ha condotto, nel febbraio 2003, uno studio di fattibilità riguardante il software OS. Sulla base di tale studio è stata sancita la pari dignità tra software proprietario e OS negli acquisti della pubblica amministrazione, al fine di garantire una migliore competizione nel mercato.

La stessa agenzia ha, tra l'altro, condotto e pubblicato un test di interoperabilità tra StarOffice Writer 6.0 and Office Word 2003 Beta 2.

## **1.4.3. Paesi extra UE**

### **Australia**

Il NOIE (National Office for the Information Economy) mantiene un sito web<sup>26</sup> di risorse open source.

### **Brasile**

Il presidente brasiliano Lula da Silva sta conducendo una politica per introdurre l'obbligo di adottare software open source (in particolare Linux) nei Ministeri, agenzie e imprese statali per quanto riguarda i nuovi computer installati. In particolare l'obiettivo è che, nel corso del 2004, almeno l'80% dei computer governativi acquistati utilizzino software open source .

---

<sup>24</sup> <http://www.e-extremadura.org/>

<sup>25</sup> <http://www.statkontoret.se/english/index.htm>

<sup>26</sup> [http://www.govonline.gov.au/projects/egovernment/Better\\_Infrastructure/OSS/oss\\_resource\\_page.htm](http://www.govonline.gov.au/projects/egovernment/Better_Infrastructure/OSS/oss_resource_page.htm)

Nell'ottobre 2003 il governo federale brasiliano ha pubblicato il documento "Free software Implementation Guidelines". Il governo federale brasiliano ha un portale dedicato al software libero e open source<sup>27</sup>.

### **Cina, Giappone e Sud Corea**

Nel settembre 2003 i governi dei tre Paesi hanno annunciato che coopereranno col settore privato al fine di sviluppare nuovi sistemi operativi per dispositivi quali i cellulari, fotocamere digitali e sistemi di navigazione per auto, nonché per il mercato dei server di rete. Tali implementazioni saranno basate su Linux (fonte: Dravis Group, 2003).

La Cina sta sostenendo il software libero con il supporto a distribuzioni nazionali, in particolare Elx Linux<sup>28</sup>, e con l'installazione di GNU/Linux su desktop a basso costo al posto dei sistemi Microsoft.

### **India**

Il governo indiano ha lanciato la "Linux India Initiative" che consiste nello sviluppo di centri di risorse, gruppi di interesse e progetti pilota al fine di supportare la localizzazione di software open source e lo sviluppo di ricerche.

### **Sud Africa**

Nel gennaio 2003 il "Government Information Technology Officers' Council" (GITOC) ha pubblicato il documento "Using Open Source Software in the South African Government<sup>29</sup>" in cui si propone un chiaro insieme di raccomandazioni riguardanti la politica del governo del Sud Africa per il software OS.

Si cita anche: "SA Action Plan: Open Software & Open Standards in South Africa - A Critical Issue for Addressing the Digital Divide - January 2002<sup>30</sup>".

### **USA**

*The Center of Open Source & Government*<sup>31</sup> è un centro che collabora in generale con vari governi (e in particolare con quello USA) per quanto concerne le politiche e le strategie relative all'open source.

Nell'ottobre 2000 il "President's Information Technology Advisory Committee (PITAC)" ha prodotto il rapporto "Developing Open Source Software to Advance High End Computing<sup>32</sup>". In questo documento si raccomanda l'utilizzo di metodologie open source nello sviluppo di progetti di "high end computing".

La MITRE Corporation ha intrapreso uno studio<sup>33</sup> per determinare quali applicazioni OS sono utilizzate dal DoD (Dipartimento della Difesa), come vengono utilizzate, le implicazioni di sicurezza e di affidabilità, e l'impatto delle licenze OS sul software proprietario. Lo studio è stato pubblicato nell'ottobre 2002. Le principali conclusioni sono state che il software OS gioca un ruolo molto più critico di quanto preconizzato, e

<sup>27</sup> <http://www.softwarelivre.gov.br>

<sup>28</sup> <http://www.elxlinux.com>

<sup>29</sup> [http://www.oss.gov.za/docs/OSS\\_Strategy\\_v3.pdf](http://www.oss.gov.za/docs/OSS_Strategy_v3.pdf)

<sup>30</sup> <http://www.naci.org.za/docs/opensource.html>

<sup>31</sup> <http://www.egovos.org/>

<sup>32</sup> <http://www.ccic.gov/pubs/pitac/pres-oss-11sep00.pdf>

<sup>33</sup> <http://www.egovos.org/pdf/dodfoss.pdf>

che le applicazioni più importanti sono pertinenti alle seguenti aree: supporto all'infrastruttura, sviluppo software, sicurezza e ricerca. Un aspetto inatteso è stato il grado di dipendenza della sicurezza dal software open source.

È dell'aprile 2003 il rapporto interno della Nasa “Developing an Open Source Option for NASA Software<sup>34</sup>”. In esso si afferma che l'utilizzo di software open source determina tre principali benefici:

- migliore qualità nello sviluppo del software;
- miglioramento della collaborazione;
- disseminazione (del software ) più efficace ed efficiente.

Inoltre vari stati USA (Oregon, Texas, Oklahoma) hanno introdotto leggi che, in vari modi, impongono alle agenzie statali di considerare l'uso di software open source o comunque che, nelle nuove acquisizioni di software, sia richiesto l'accesso al codice sorgente (fonte: Dravis Group, 2003).

### **Case Studies nelle nazioni in via di sviluppo (fonte: Dravis Group, 2003)**

In particolare si segnalano i seguenti casi di utilizzo di software open source:

- San Paolo, Brasile – Telecenter Project. La città di San Paolo sta implementando telecentri per fornire uso del computer e accesso Internet alle fasce emarginate di cittadini.
- Tagikistan. Il progetto consiste nella localizzazione in tagiko del desktop OS e di sviluppare materiale educativo per la scuola secondaria.
- Laos – Jahi Computer. Implementazione di un sistema per effettuare chiamate locali e oltremare via internet da fornire al villaggio di Phon Kham e altri quattro villaggi della regione.
- Goa (India). School computer Project. Il progetto prevede di consegnare PC riciclati con sistema operativo Linux a 125 scuole della regione.

## **1.5. Il modello organizzativo e i processi produttivi dell'OS**

### **1.5.1. Introduzione**

Le organizzazioni che adottano l'approccio OS si differenziano dalle aziende che sviluppano software proprietario non solo per il modello economico, ma anche e soprattutto per il modello organizzativo e di sviluppo.

Nello sviluppo di software OS, contrariamente a quanto si possa immaginare, la metodologia alla base delle attività di sviluppo, unitamente agli strumenti che ne facilitano l'implementazione, è un fattore critico per il successo del progetto. Infatti le comunità del mondo OS, poco gerarchizzate e distribuite, sia nello spazio che nel tempo, hanno bisogno di regole precise e fanno ampio uso della tecnologia per poter interagire efficacemente. In un'azienda “virtuale” di software non si tengono riunioni di lavoro, e

---

<sup>34</sup> <http://www.nas.nasa.gov/Research/Reports/Techreports/2003/nas-03-009-abstract.html>

quindi lo scambio di informazioni non può avvenire in maniera informale.

Le modalità di sviluppo di un prodotto software OS sono molto differenti da quelle generalmente adottate per il software tradizionale: i modelli classici che definiscono il Ciclo di Vita sono difficilmente applicabili a questa realtà.

La genesi e l'evoluzione di un prodotto OS sono infatti frutto di un'attività collaborativa in cui gli attori interagiscono a distanza e quasi esclusivamente senza incontri formali, e dove lo scambio di e-mail è il principale mezzo di comunicazione della comunità di sviluppo.

Nei progetti OS non esiste un modello di sviluppo definito a-priori e universalmente adottato. La comprensione delle best practice costituisce un passaggio fondamentale per la definizione di una metodologia di qualificazione e valutazione. Dà inoltre l'opportunità di partecipare attivamente e con maggiore consapevolezza al processo di formazione delle soluzioni tecnologiche, contribuendo a ridurre il gap che tradizionalmente caratterizza il rapporto tra committente e fornitore.

### **1.5.2.Organizzazione dei team di progetto**

La strutturazione decentralizzata, virtuale e collaborativa, delle comunità che producono OS, può portare a immaginare che esse non abbiano una vera organizzazione.

Al contrario, tutti i partecipanti all'interno di una comunità di sviluppo hanno generalmente ruoli ben definiti, ad esempio sviluppatore del core o di uno specifico modulo, amministratore dell'archivio del codice sorgente, revisore, utente finale. Ogni partecipante contribuisce alla comunità attraverso contenuti che possono essere:

- parti di codice sorgente,
- script,
- revisioni,
- commenti,
- bug report,
- documentazione utente o tecnica.

e comunicano tra loro attraverso:

- forum di discussione,
- mailing list,
- newsgroup.

Le comunità più consolidate, attive su più linee progettuali e con un ampio numero di collaboratori, hanno poi delle strutture organizzative articolate. Ad esse sono affidate non solo le responsabilità del processo produttivo, ma anche quei compiti tipicamente gestionali, che fatalmente emergono al crescere delle dimensioni della struttura. Il processo decisionale è anch'esso chiaramente definito e tracciabile, seppure non eseguito

nell'ambito di figure organizzative piramidali.

### **1.5.3. Modello organizzativo: il caso Apache Software Foundation.**

Il numero di progetti gestiti, la qualità dei propri prodotti e non ultimo la penetrazione di mercato che alcuni di essi hanno raggiunto, fanno di Apache Software Foundation (ASF) un caso di studio e un benchmark per le organizzazioni di tipo OS.

ASF governa tutti i progetti che portano il nome Apache attraverso una struttura evoluta con al vertice un Board of Director (BoD), responsabile della gestione degli affari generali della fondazione, incluse le questioni relative alla proprietà intellettuale, al finanziamento delle attività e all'allocazione delle risorse comuni sui singoli progetti. Il BoD è eletto dai membri della ASF. I membri della ASF sono nominati per invito degli altri membri, in base a valutazioni di merito sulle collaborazioni prestate alla ASF. Sotto questo primo livello manageriale si entra subito nell'organizzazione operativa.

La leadership di progetto è affidata al Project Management Committee (PMC) di cui fanno parte un gruppo di sviluppatori che decidono cosa è distribuito come prodotto, gestiscono le risorse condivise, risolvono questioni relative a dispute sulle licenze. Al PMC si accede per invito di uno o più membri del PMC e con il consenso di tutti gli altri. Il PMC stabilisce, infine, le regole da osservare nell'ambito del progetto.

Il livello organizzativo successivo è quello dei Server Committers, anch'esso composto da un gruppo di volontari che sono responsabili degli aspetti tecnici del progetto. Questo gruppo ha diritto di scrittura sull'archivio dei sorgenti, e ha diritto di voto su questioni tecniche. Nuovi appartenenti possono aggiungersi solo per invito degli appartenenti al PMC e previo consenso degli altri componenti.

Il terzo livello è costituito dagli sviluppatori (Apache Developers), volontari che contribuiscono con codice, documentazione o altre risorse al progetto. Generalmente uno sviluppatore che fornisce contributi di valore per circa sei mesi è invitato a entrare nella squadra dei Committers.

### **1.5.4. Gestione delle risorse umane.**

Quella delle comunità OS è una realtà fortemente virtualizzata, e naturalmente anche i rapporti di lavoro sono, nella maggior parte dei casi, sostanzialmente informali. Quindi parlare di Gestione delle Risorse Umane sembra un paradosso, eppure è lecito pensare che nei processi di formazione dei team di sviluppo e dei diversi ruoli organizzativi, come visto ad esempio nel caso ASF, si inneschino meccanismi impliciti di selezione (inviti a far parte del team), di formazione (per esempio peer review), di motivazione (auto selezione dei compiti) e di incentivazione (gratificazione dalla comunità).

Se confrontiamo tutto ciò con la realtà delle aziende legate al processo convenzionale di assunzione dei collaboratori, ci dobbiamo aspettare che il livello qualitativo nelle comunità sia decisamente elevato. Lo stesso meccanismo di assegnazione dei compiti, sostanzialmente autoelettivo, può rivelarsi più efficace dal punto di vista della motivazione e in definitiva della produttività, se confrontato con un meccanismo di assegnazione per via gerarchica.

### 1.5.5. Modello di sviluppo.

Il meccanismo principale per lo sviluppo e l'evoluzione di un software OS è il cosiddetto “software extension mechanism”, un sistema, a volte regolato da specifiche licenze (ad esempio GPL), che garantisce e a volte vincola la disponibilità del codice sorgente, e che inoltre favorisce lo sviluppo incrementale.

Le comunità OS non adottano esplicitamente le moderne tecniche di ingegneria del software, ma lo sviluppo evolve attraverso processi paralleli e concorrenti, in cui il singolo processo può adottare un autonomo modello di sviluppo.

Nei progetti OS la necessità di operare in un certo modo ha determinato un nuovo approccio nelle modalità operative di sviluppo del software, che successivamente sono state formalizzate in tutte quelle metodologie definite “agili”, cioè basate sulla prototipazione, sulla produzione frequente di versioni e sull'evoluzione adattativa.

### 1.5.6. Genesi del Progetto.

I meccanismi più frequenti che generano un progetto sono:

**Proposizione degli Autori:** gli ideatori di un progetto realizzano un primo prototipo di prodotto, utilizzando il più possibile componenti software già realizzate da altri progetti OS e facendo riferimento a standard applicabili al progetto. Solo se la comunità degli sviluppatori troverà la proposta interessante il team di progetto comincerà a prendere forma e consistenza numerica. In questo caso il progetto avrà delle buone probabilità di giungere ad un primo rilascio, la fatidica versione 1.

**Contributo dall'industria:** esistono molteplici esempi di generazione di progetti OS originati da contributi da parte dell'industria. Talvolta la decisione di rendere open source un prodotto viene presa da piccole aziende, non più in grado di sostenere i costi di ricerca e sviluppo, marketing e vendita del prodotto. Talvolta sono i colossi come IBM e Sun Microsystems, che intraprendono questa strada, per motivazioni che non sono altrettanto trasparenti, ma che probabilmente si possono accreditare sia alla necessità di utilizzare un paradigma di sviluppo più agile che ad un cambio strategico volto a privilegiare l'offerta di servizi piuttosto che l'offerta di prodotti.

**Iniziativa del Committente:** in questo approccio, che comincia a manifestare i primi esempi solo da poco tempo, è il committente che chiede ai fornitori di avvalersi di un modello di sviluppo OS.

### Il caso OpenOffice.org.

Un caso eclatante è stato il lancio del progetto OpenOffice.org. Spinta da motivazioni strategiche e commerciali, la Sun Microsystems, acquisì, sul finire degli anni 90, un'azienda tedesca (la Star Division) proprietaria della suite di produttività personale StarOffice. Fu ben presto evidente ai manager di Sun che l'impresa di contendere significative fette di mercato a Microsoft Office fosse ben più difficile e onerosa del previsto e così il progetto StarOffice venne accantonato. Il progetto fu tuttavia rilanciato più tardi attraverso la pubblicazione del codice sorgente del prodotto e l'affidamento dello stesso alla comunità OS, con il contributo degli sviluppatori già in forza alla divisione StarOffice di Sun. Questa è stata indubbiamente una scelta di successo poiché oggi OpenOffice.org conta migliaia di sviluppatori che contribuiscono in forme e misure



diverse a realizzare e supportare un prodotto che conta installazioni nell'ordine dei milioni ed è stato tradotto in 25 lingue diverse. Tra le molteplici ragioni di questo successo, vi è certamente il modello di sviluppo adottato, in grado di produrre ricadute positive sulla economicità ed efficienza del prodotto, lasciandolo nel contempo libero da qualsiasi vincolo sui formati utilizzati per la manipolazione e memorizzazione dei dati.

### **Un caso emblematico: EIONET.**

EIONET è una rete di agenzie europee finalizzata all'osservazione dei fenomeni ambientali e alla disseminazione delle informazioni prodotte. Il principale canale informativi usato da EIONET è il suo portale (<http://www.eionet.eu.int/>), la cui prima realizzazione fu affidata ad una azienda attraverso un tradizionale contratto di sviluppo di tipo custom. Nel tempo, questo rapporto contrattuale ha portato EIONET alla identificazione di una serie di criticità bloccanti quali:

- Unico interlocutore possibile per nuovi sviluppi funzionali.
- Incapacità di influire sui tempi e sui modi di esecuzione dei nuovi sviluppi.
- Scarso potere contrattuale nella negoziazione dei prezzi.
- Incapacità di fornire un servizio ai clienti indipendente dal fornitore.
- Incapacità di rilevare in maniera trasparente le cause di malfunzionamento del sistema.

EIONET ha quindi cambiato il proprio approccio alla produzione del software, sia sviluppato in casa che commissionato a terzi, rilasciando ogni cosa come software open source. I benefici che EIONET ha rilevato da questo cambio di strategia sono:

- Capacità di negoziare i prezzi.
- Possibilità per altre società di contribuire agli sviluppi o di fornire supporto.
- Protezione degli investimenti contro la scomparsa del fornitore.
- Accresciute possibilità per il riuso applicativo.

### **1.5.7. Pianificazione di prodotto.**

Nelle organizzazioni OS, la forte apertura alla contribuzione pluralistica (utenti, sviluppatori, ricercatori, etc.) e la trasparenza informativa sembrano produrre una base più ampia per la costruzione delle caratteristiche funzionali e tecniche dei prodotti e anche una migliore percezione delle priorità.

Nel modello tradizionale infatti, la responsabilità della pianificazione è maggiormente concentrata, tipicamente riservata al Product Manager e al marketing di prodotto, e considerata un'informazione riservata da cui dipende, in parte rilevante, la competitività del produttore.

Anche se nei progetti OS viene concordata e pubblicata una roadmap di prodotto con la

pianificazione di massima delle versioni e della loro futura evoluzione, non sempre è definita la tempificazione, ovvero quali funzionalità verranno implementate nelle singole versioni.

### **1.5.8. Analisi dei requisiti e specifiche**

L'analisi dei requisiti non è un'attività di progettazione formale eseguita a partire dalle specifiche emesse da un committente. I requisiti, nei progetti OS, emergono e si manifestano nell'ambito delle mailing list e dei forum di discussione. Alcune volte i requisiti vengono addirittura formalizzati alla fine dello sviluppo: in questo caso si parla di “retrospective requirements specification”.

Ciò avviene perchè, come detto precedentemente, le funzionalità vengono aggiunte al prodotto man mano che sono rese disponibili dagli sviluppatori in forza alla comunità.

### **1.5.9. Strumenti di sviluppo**

C++ e Java sono i linguaggi più utilizzati nei progetti OS, e incidono sulla tipologia di strumenti di sviluppo adottati specialmente per la generazione dell'eseguibile.

In particolare possiamo distinguere gli strumenti a supporto dello sviluppo di applicazioni scritte in linguaggi C e C++, che richiedono il porting su specifica piattaforma, e quelle in Java, che sono multiplatforma.

Nel primo caso è predominante l'utilizzo della triade *gcc*, *make* e *configure*, mentre nel secondo caso gli strumenti più diffusi sono JDK e *ant*. Quest'ultimo, basato su XML, è in fase di crescente diffusione come alternativa a *make*.

Nella comunità OS, a oggi non hanno trovato molta fortuna gli strumenti di supporto al design, probabilmente perchè tutta l'enfasi è posta sul codice sorgente come contenitore della conoscenza necessaria alla comprensione del sistema.

Infatti, specialmente nelle applicazioni sviluppate in Java, si tende a concentrare anche la parte di documentazione e configurazione all'interno del codice sorgente, utilizzando strumenti come Javadoc o il più potente Xdoclet.

Anche sul fronte degli strumenti IDE esiste una differenziazione tra C, C++ e Java, che però ha più ragioni storiche che altro. Infatti è facile trovare progetti OS scritti in Java che fanno uso di strumenti IDE come Eclipse, utilizzati per gestire in modo automatico (e di conseguenza in modo comune a tutta la comunità di sviluppatori) le operazioni di build, debug, test, refactoring, generazione automatica della documentazione e dei file di descrizione/configurazione.

### **1.5.10. Controllo di versione, system build e rilascio incrementale**

La manutenzione del software, cioè l'aggiunta o l'eliminazione di funzionalità, il debugging, il tuning, il porting su piattaforme differenti è un processo molto ricorrente nell'ambito delle comunità di sviluppo del software OS. A differenza del software tradizionale, le versioni del software OS evolvono attraverso cambiamenti spesso minimi ma frequenti.

Nel modello OS i rilasci avvengono con grande frequenza, secondo il paradigma “release

early, release often”, a vantaggio della capacità di gestire feedback su funzionalità, affidabilità, usabilità, prestazioni e quant'altro può emergere dalla comunità attiva sul progetto.

Una pratica comune a tutti i progetti OS è quella di gestire almeno tre tipi di rilascio:

- Stable Release: indica il livello qualitativo più elevato (il prodotto è adatto all'utilizzo quotidiano in ambiente operativo);
- Nightly Release: sono prodotte a ciclo continuo per diffondere gli ultimi aggiornamenti appena resi disponibili dagli sviluppatori;
- Development Release: sono prodotte a distanza di poche settimane una dall'altra, a partire dalle primissime fasi del progetto, immediatamente a ridosso del kick-off.

Un'ulteriore convenzione, molto interessante, adottata nei rilasci OS, è la numerazione: tutti i progetti in fase di avvio rilasciano secondo una notazione pre-1.0 o 0.x, che indica che il prodotto è instabile, implementa un insieme basilare di funzionalità ed è gestito dagli ideatori, in molti casi un singolo sviluppatore. Una rapida evoluzione delle versioni e il crearsi di una comunità saranno gli indicatori di un progetto destinato a durare.

Il controllo di configurazione è quasi sempre basato su CVS (i siti come sourceForge lo usano come archivio del prodotto).

### 1.5.11.Documentazione

Nel ciclo di vita di un prodotto OS vengono prodotte una quantità enorme di informazioni, che però non sono sempre organizzate in modo strutturato o comunque in maniera tradizionale. Infatti la distinzione tra documentazione di sviluppo, documentazione tecnica e documentazione utente non sempre è così netta.

La qualità e la completezza della documentazione va di pari passo con lo stato evolutivo del progetto e con l'ampiezza della base collaborativa. Nuovi progetti, o progetti poco maturi, hanno tipicamente poca documentazione, che si limita al README, allo HOWTO e all'INSTALL, mentre progetti che nel tempo hanno raggiunto grande popolarità sono generalmente arricchiti da una grossa mole di documentazione, a volte veri e propri testi di riferimento, spesso disponibili anche in diverse lingue.

La documentazione prodotta può essere classificata in tre tipologie.

- Documentazione Minimale, che comprende i seguenti documenti:
  - *README*. Documento di base, quasi sempre allegato al codice sorgente. Fornisce il minimo indispensabile sulle funzionalità e sull'uso del prodotto.
  - *HOWTO*. Documentazione tecnica e utente che ha come scopo finale la descrizione degli usi più comuni del prodotto (appunto “how to...”). Gli howto sono spesso disponibili on-line.
  - *INSTALL*. Documentazione con le istruzioni per l'installazione del prodotto a partire dalla versione sorgente in formato compresso (.tar, .gz o .tgz).

- *FAQ*. Documentazione prevalentemente tecnica che, come suggerisce il termine Frequently Asked Questions, descrive “soluzioni” alle problematiche più comuni inerenti il prodotto.
- Documentazione Utente. La tipologia di documentazione per l'utente finale è strettamente legata al tipo di prodotto. Ad esempio, la suite di documenti per un prodotto come Mozilla è molto differente da quella di MySQL. Di seguito viene riportata una lista parziale:
  - *Help Pages*. Documentazione spesso inclusa nel prodotto o disponibile online. Alcune volte, quando inclusa nel prodotto, può essere arricchita da strumenti quali “Help Agent” o “Assistenti”, cioè software che facilitano la consultazione immediata delle risorse documentali.
  - *Getting Started Guide*. Documento introduttivo che permette a chi non ha mai utilizzato il prodotto di avere una visione generale sulle caratteristiche del prodotto e sul modo più veloce di utilizzarlo. Talvolta questo documento è sostituito dal Tutorial.
  - *Reference Manual*. Documento dettagliato sul prodotto e sulle sue funzionalità.
  - *Administrator's Guide*. Documentazione per l'amministratore.
  - *Developer's Guide*. Tradotto: “Guida dello sviluppatore”, da non confondere con la documentazione per lo sviluppo del prodotto. Questo documento, presente solo per prodotti di sviluppo (es. Jboss, Postgres, MySQL) contiene le informazioni necessarie a chi deve sviluppare un'applicazione utilizzando lo specifico prodotto.
  - *Descrizione delle API*. Documentazione tecnica e dedicata alla integrazione del prodotto con sistemi esterni. Esistono strumenti specifici che facilitano la produzione di tale documentazione, e strumenti di nuova generazione come Javadoc o Xdoclet che ne permettono la creazione automatica a partire da informazioni che lo sviluppatore include direttamente nel codice sorgente.
- Documentazione per lo Sviluppo. Questo tipo di documentazione è rivolta a chi vuole unirsi al gruppo di lavoro che sviluppa il prodotto. Ogni progetto realizza un proprio insieme di documentazione a supporto degli sviluppatori più o meno completo. I progetti maturi hanno in genere una:
  - *Styleguide*. Documentazione di supporto allo sviluppo che delinea le regole di scrittura del codice sorgente (indentazione, convenzioni sui nomi, ecc.) e in alcuni casi include indicazione per la gestione della localizzazione e della portabilità.

È importante che un progetto OS disponga di uno strumento di divulgazione che faciliti la condivisione di tali informazioni. Generalmente, dopo una prima fase in cui si utilizzano gli strumenti forniti da ambienti come SourceForge, viene creato un portale dedicato al progetto che contiene diverse aree informative come documentazione di progetto, forum di discussione, link agli strumenti di gestione della versione, link agli strumenti di gestione dei bug, link ai documenti relativi agli standard di riferimento adottati nel

progetto.

È importante notare che la documentazione relativa a progetti OS spesso dà origine ad interessanti progetti editoriali che rappresentano un importante indotto economico, legato al software OS, che direttamente o indirettamente sostiene e aiuta la comunità. Ad esempio, lo stesso sito web che ospita il progetto OS propone a volte documentazione da acquistare on-line: gli autori di tali volumi sono molto spesso sviluppatori, collaboratori, integratori e utilizzatori o comunque persone legate allo sviluppo del progetto stesso.

È significativo citare il caso di Jboss Group, che ha fatto una scelta precisa per quanto riguarda la documentazione. Per evitare la proliferazione di documenti di bassa qualità e spingere gli sviluppatori a scrivere documentazione, ha messo a disposizione gratuita solo due documenti: *Getting Started 3.2.x*<sup>35</sup> e *JBoss 3.2 Workbook and Examples*<sup>36</sup>, creando una collezione di manuali a pagamento i cui proventi vanno a chi sviluppa tale documentazione.

Si cita inoltre il caso di Linux, la cui documentazione è disponibile in lingua inglese, francese, spagnolo ed altre ancora grazie al coordinamento del “*The Linux Documentation Project*” (LDP)<sup>37</sup>. Tale documentazione è anche disponibile in italiano attraverso l’*Italian Linux Documentation Project*<sup>38</sup> (ILDP), che si pone l’obiettivo di aiutare la diffusione di GNU/Linux sia tramite la produzione di documentazione, sia tramite la traduzione della documentazione originale in inglese.

### 1.5.12.Test

Nel mondo OS, ogni singolo progetto adotta un insieme di criteri propri per il rilascio di una versione del prodotto. Tali criteri si basano soprattutto sull'accettazione da parte della comunità degli sviluppatori di versioni preliminari, spesso indicate con il nome di “*pre-release*” o “*release-candidate*”.

L'attività di test è quindi tutta a carico della comunità stessa, che favorisce e sollecita i partecipanti a collaborare alla verifica delle pre-release e a contribuire con numerosi feedback.

Tipicamente la presenza di una nuova release-candidate è annunciata dalla comunità degli sviluppatori attraverso la mailing list del progetto, o tramite gli altri strumenti di comunicazione a disposizione. Ogni sviluppatore è libero di condurre i test nel modo che più ritiene appropriato, ma con il vincolo di dare visibilità dei risultati ottenuti. Dopo un certo periodo che una pre-release è stata utilizzata, analizzata e corretta, la comunità la accetta e quindi può diventare una stable-release.

I test effettuati dalla comunità sono in genere soprattutto:

- “Smoke Test”. Sono una serie di semplici e parziali test sul prodotto che hanno come scopo la verifica del funzionamento di base del prodotto, e consistono nel controllare che i pattern di utilizzo più comune abbiano un comportamento corretto e senza bug.

---

<sup>35</sup> <http://www.jboss.org/modules/html/docs/jbossj2ee.pdf>

<sup>36</sup> <http://www.oreilly.com/catalog/entjbeans3/workbooks/ejbwJBoss.Files.zip>

<sup>37</sup> <http://www.tldp.org/>

<sup>38</sup> <http://www.pluto.linux.it/ildp/>

- “Regression Test”. Questa tipologia di test ha lo scopo di verificare che, l'aggiunta di una nuova funzionalità non incida sul corretto funzionamento delle funzionalità già presenti nel prodotto.

Questo modello di test garantisce ottimi risultati, che vanno a incidere in modo positivo sulla qualità del prodotto. Infatti l'ampio numero di collaboratori che si dedica “indirettamente” ad attività di tester è molto superiore a quella dei prodotti tradizionali, e quindi il numero di test effettuati, seppur non rigidamente formalizzati, permette una dettagliata validazione della versione di prodotto secondo uno spettro di copertura molto esteso.

### **1.5.13. Supporto e manutenzione evolutiva**

Questa è una caratteristica critica del software OS, nel senso che è stata spesso citata come un punto di debolezza in questo modello di sviluppo.

Tutte le organizzazioni osservate utilizzano sistemi on-line (bugzilla) per la segnalazione di problemi relativi al funzionamento dei prodotti. Il sistema mette a disposizione operazioni di ricerca, che consentono di verificare se il problema è già noto e quale sia il suo stato corrente (assegnato, aperto, chiuso, ecc.).

Vi sono anche i classici strumenti di mailing list e chat per chiedere informazioni alla comunità degli utenti e degli sviluppatori. È anche possibile risolvere il problema attraverso l'iniziativa diretta, ovvero accedendo al codice sorgente per apportare le necessarie correzioni. In questo caso esistono delle procedure ben definite per consolidare le correzioni (patch) all'interno della distribuzione ufficiale del prodotto, in modo che tutti gli utilizzatori ne possano beneficiare.

A questa tradizionale modalità di erogazione del supporto si sta affiancando un nuovo modo di concepire il supporto attraverso la nascita di “open company”, che rappresentano il lato commerciale di un prodotto open source.

La prima società di questo tipo è stata RedHat, alla quale si sono aggiunte società come MySQL AB e Jboss Group Inc. Tali società offrono a pagamento servizi di supporto che vanno dalla semplice chiamata telefonica alla fornitura di servizi professionali specializzati e alla garanzia sul prodotto.

## **1.6. Scenario normativo**

### **1.6.1. Il Diritto d'Autore e il software OS**

La legge sul Diritto d'Autore tutela esplicitamente il software come “opera dell'ingegno umano”, essendo evidente lo sforzo creativo necessario per disegnare e realizzare (sviluppare) un'applicazione software.

La legge sulla protezione del diritto d'autore concernente le opere originali e creative riconosce due possibili prerogative per esercitare tale “diritto”:

- Diritti morali (paternità dell'opera - non cedibile a terzi).

- Diritti patrimoniali (per lo sfruttamento economico dell'opera. Possono essere ceduti ad altri soggetti concedendo a questi la facoltà di utilizzare l'opera medesima, con licenza d'uso, in cambio di un compenso economico).

È esclusiva facoltà dell'autore decidere le modalità di circolazione della propria opera. Nel caso del software, l'autore può indicare le condizioni d'uso autorizzate. Ciò viene specificato nella "licenza d'uso", un contratto con valore legale, allegato al programma, che indica se e in che modo tale opera possa essere utilizzata, distribuita e modificata.

Alla licenza d'uso si può accompagnare:

- il solo programma eseguibile ("codice oggetto", cioè il codice compilato, in forma binaria, eseguibile dal calcolatore, ma non comprensibile per gli esseri umani);
- il programma eseguibile e i "sorgenti" (il listato della codifica, cioè il codice espresso in un linguaggio di programmazione comprensibile per gli addetti ai lavori - i programmatori -, utilizzato per descrivere il flusso delle operazioni che un programma deve eseguire).

Si evidenzia che la disponibilità dei sorgenti, di per sé, non esprime la garanzia della ispezionabilità del codice, ovvero del suo studio o della sua modifica o distribuzione. Ciò perché la licenza d'uso potrebbe esplicitamente vietare tali utilizzi, fornendo l'accesso ai sorgenti unicamente per determinati scopi (ad esempio, consentire la verifica dell'assenza di "backdoor"), ovvero in specifiche circostanze (ad esempio, al verificarsi di anomalie di una certa gravità, o in caso di fallimento del fornitore) e molto spesso rispettando obblighi ben definiti a priori (ad esempio, rispettando la clausola NDA, "non disclosure agreement").

Si osservi inoltre che l'assenza di una qualunque licenza software, stando alla convenzione di Berna sul diritto d'autore, implica che il software non possa essere distribuito né modificato senza l'esplicito consenso da parte dell'autore, quindi il cosiddetto software di "pubblico dominio" non può essere considerato in alcun modo software open source.

Quanto detto aiuta a identificare la terminologia da utilizzare e quindi a chiarire il problema del rapporto tra legge sul diritto d'autore e software open source. Infatti, parlare di software libero altro non significa che riferirsi a una particolare norma per l'attuazione (esercizio) di uno dei diritti che la legge medesima accorda all'autore, ovvero le modalità di distribuzione delle opere dell'ingegno.

Il software open source, in quanto opera dell'ingegno, rientra nella legge sul Diritto d'Autore, ed è regolato e disciplinato dalla Legge n. 633 del 1941<sup>39</sup> e successive modifiche introdotte dal decreto legislativo n. 68 del 2003<sup>40</sup>. Infatti, la disponibilità e le

<sup>39</sup> Legge 22 aprile 1941 n. 633 - Protezione del diritto d'autore e di altri diritti connessi al suo esercizio (G.U. n.166 del 16 luglio 1941). L'Art. 64-ter della l. 633/41 afferma che non è possibile vietare all'utente finale di effettuare una copia di riserva del software legittimamente acquisito. L'Art. 64-quater afferma che "L'autorizzazione del titolare dei diritti non è richiesta qualora la riproduzione del codice del programma di elaboratore e la traduzione della sua forma ai sensi dell'art. 64-bis, lettere a) e b), compiute al fine di modificare la forma del codice, siano indispensabili per ottenere le informazioni necessarie per conseguire l'interoperabilità, con altri programmi, di un programma per elaboratore creato autonomamente" Purché siano soddisfatte alcune condizioni, quali: mantenere la riservatezza delle informazioni di cui si è venuti in possesso e non fare uso improprio di tali informazioni.

<sup>40</sup> Decreto legislativo 9 aprile 2003, n. 68 - Attuazione della direttiva 2001/29/CE sull'armonizzazione di taluni aspetti del diritto d'autore e dei diritti connessi nella società dell'informazione (G. U. n. 87 del 14.4.2003 - S. O.

regole di utilizzo del software sono riferibili a una precisa scelta dell'autore, che stabilisce le modalità sulla circolazione della sua opera.

A seguito della diffusione del software OS, si è reso necessario definire nuove direttive per regolare la diffusione di questo tipo di prodotti. La fonte primaria per la regolamentazione del diritto d'autore è la succitata Legge 633/41, su cui è intervenuto specificamente, per quanto riguarda il software, il Provvedimento Comunitario 91/250/CE<sup>41</sup> sulla tutela giuridica del software, modificata dalla Direttiva 93/98/CEE<sup>42</sup> del Consiglio; e ai sensi della legge sul diritto d'autore e il decreto legislativo 518/92<sup>43</sup> di recepimento (modificata dalla Legge n. 248/00<sup>44</sup>).

L'attuazione della legge si pone come obiettivo di definire, in particolare, la tutela del software e quindi, identificare e tutelare i molteplici interessi degli utenti e dei fornitori dell'industria informatica, in materia di circolazione dei diritti e di regime delle esclusive in materia di software.

Il software open source è dunque compatibile con la legislazione italiana, che già prevede la protezione del diritto d'autore, vale a dire la tutela del segreto e della proprietà intellettuale frutto del lavoro di aziende o di singoli professionisti.

La differenza tra il software open source e il software proprietario non riguarda quindi la protezione del diritto d'autore, ma riguarda solo ed esclusivamente le modalità di rilascio dello stesso all'utente. Nel primo caso si rilascia sempre all'utente anche il codice sorgente e alcuni diritti che l'autore concede anche a terzi, non riservandosi in maniera esclusiva; mentre nel secondo caso si rilascia solo il programma eseguibile.

Quindi il software open source consente all'utente un maggior controllo sui lavori svolti dal fornitore di software, mentre, con il software proprietario questo controllo non è possibile, a meno di accordi specifici con il fornitore che autorizzino l'utente ad accedere al codice sorgente.

### **1.6.2. Definizioni e aspetti giuridici del modello OS**

È necessario prestare attenzione alle definizioni tecniche utilizzate per identificare ogni tipo di software open source. Ciò consente di inquadrare gli aspetti giuridici riguardanti il diritto d'autore che si devono rispettare per l'uso di quello specifico prodotto open source.

Le principali definizioni sono:

---

n.61) - Testo in vigore dal 29.4.2003.

<sup>41</sup> PROVVEDIMENTO COMUNITARIO - Direttiva 91/250/CEE del Consiglio, del 14 maggio 1991, relativa alla protezione legale dei programmi per elaboratore.

<sup>42</sup> Direttiva 93/98/CEE del Consiglio, del 29 ottobre 1993, concernente l'armonizzazione della durata di protezione del diritto d'autore e di alcuni diritti connessi

<sup>43</sup> Decreto legislativo 29 dicembre 1992 n. 518 - Attuazione della direttiva 91/250/CEE relativa alla tutela giuridica dei programmi per elaboratore. (S.ORD. ALLA G.U. N. 306 SERIE GENERALE PARTE PRIMA DEL 31 12 1992)

<sup>44</sup> Legge 18 agosto 2000, n. 248 - "Nuove norme di tutela del diritto d'autore" (Pubblicata nella Gazzetta Ufficiale n. 206 del 4 settembre 2000)



- Software Libero (Free Software). Tale dizione indica una categoria di licenze predisposte inizialmente da Richard Stallman e adottate dalla Free Software Foundation<sup>45</sup>. Le caratteristiche principali di questa modalità di distribuzione del software sono descritte nella definizione di Free Software<sup>46</sup>, e garantiscono all'utente finale le seguenti tutele:

- *permesso di utilizzo, per qualunque scopo;*
- *permesso di analizzare il codice sorgente ed adattarlo alle proprie necessità;*
- *permesso di redistribuirlo, eventualmente modificato.*

In particolare, tra le licenze libere esiste un sottoinsieme di licenze, denominate “copyleft” (permesso d'autore), che non consentono all'utente di porre ulteriori restrizioni all'uso, di fatto impedendo che l'utente possa arbitrariamente limitare i diritti concessi dall'autore originale (clausola di persistenza). Quindi, tra le varie conseguenze dell'adozione di licenze copyleft, quali ad esempio la GPL (General Public License - GNU GPL)<sup>47</sup> c'è l'impossibilità di “chiudere” (nel senso di rendere proprietaria) un'applicazione originariamente distribuita con licenze libere.

All'altro estremo dello spettro delle licenze libere si pone il modello di licenza elaborato all'Università di Berkeley per la distribuzione dei BSD<sup>48</sup> (un'altra

---

<sup>45</sup> La Free Software Foundation (FSF) è lo sponsor primario del Progetto GNU, “la missione della FSF è quella di conservare, proteggere e promuovere la libertà di utilizzare, studiare, copiare, modificare e ridistribuire software per computer, e di difendere i diritti degli utenti del software libero. La Fondazione si finanzia attraverso le donazioni, in denaro o attrezzature, da parte di privati, aziende ed altre fondazioni. Altra fonte di finanziamento è la vendita del proprio software, anche in versioni personalizzate per l'utente e di altro materiale, dai manuali ai gadget. In questo modo la FSF è riuscita a ingaggiare un gruppo di programmatori professionisti per lo sviluppo di Free Software a tempo pieno e per l'assistenza ai propri clienti”. La FSF è diventata un punto di riferimento per tutti i programmatori volontari che contribuiscono all'evoluzione del progetto GNU. Nella Free Software Directory, sono ospitati i collegamenti a 1743 progetti con licenza GNU/GPL o compatibili con il Free Software. Il software ospitato è “garantito” dalla reputazione della FSF che richiede un livello di qualità del prodotto basato su standard molto elevati.

<sup>46</sup> Per la definizione di Software Libero si rimanda al sito della Free Software Foundation: <http://www.gnu.org/philosophy/free-sw.it.html>

<sup>47</sup> La GNU General Public License è una licenza open source per software libero. Il testo ufficiale della licenza è disponibile all'URL <http://www.gnu.org/licenses/gpl.html>. Di solito, insieme ai file della distribuzione del software rilasciato con questa licenza, si trova COPYING che contiene il testo della stessa. La traduzione non ufficiale in italiano è disponibile all'URL <http://www.softwarelibero.it/gnudoc/gpl.it.txt>. È spesso indicata con l'acronimo GNU GPL o (quando non c'è il rischio di confondersi con un'altra “General Public License”) semplicemente GPL. Per evitare un errore alquanto comune, si tenga presente che GPL non significa Gnu Public License. La GNU GPL è stata scritta da Richard Stallman e Eben Moglen nel 1989, per distribuire i programmi creati dal Progetto GNU. È basata su una licenza simile usata per le prime versioni di GNU Emacs. Contrapponendosi alle licenze per software proprietario, la GNU GPL permette all'utente: libertà di utilizzo, copia, modifica e distribuzione; a partire dalla sua creazione è diventata una delle licenze per software libero più usate.

<sup>48</sup> L'acronimo BSD (Berkley Software Distribution) indica una delle varianti originarie di Unix, sviluppata presso l'università di Berkeley in California e alla base di una delle due famiglie principali di sistemi operativi liberi attualmente più diffusi, tra cui gli esponenti più noti sono OpenBSD, FreeBSD e NetBSD. L'acronimo viene usato anche per indicare la licenza BSD, una licenza libera molto semplice sotto la quale è rilasciata la maggior parte delle componenti dei sistemi citati: non copyleft (espressione inglese copyleft, gioco di parole su copyright, indica un tipo di licenza libera per la quale pur garantendo le libertà previste dalla definizione, vengono imposte delle restrizioni sul rilascio di opere derivate in modo tale da far sì che queste si mantengano sempre libere, generalmente sotto la stessa licenza dell'opera originale). Il copyleft, scardina e ribalta l'uso ed il significato corrente del copyright. Ciò che viene usualmente vietato nel copyright, qui, è permesso e difeso con i medesimi strumenti; è un impiego alternativo delle leggi sul diritto d'autore, con il risultato di ottenere un'adeguata

famiglia di sistemi Unix). Questa licenza consente di rendere “proprietaria” una soluzione software, ma nella versione originale richiede che ogni versione distribuita riporti un richiamo all’università di Berkeley (clausola pubblicitaria). Dunque è possibile utilizzare una soluzione software regolata da questa licenza, modificarla e distribuirla come si preferisce, rispettando la licenza Berkeley, applicando la GPL (solo nel caso della BSD modificata) o, al contrario, un modello rigidamente proprietario.

Il Software Libero in genere, è ceduto senza alcuna garanzia. La ragione dell’assenza di garanzia è conseguenza del fatto che l’autore originario rischia di essere chiamato in cause legali anche per modifiche non apportate da lui. Garantire questo tipo di software diventerebbe, quindi, un ostacolo alla sua diffusione ed esporrebbe l’autore a rischi legali e finanziari non prevedibili a priori. È concessa garanzia, ma limitata - come avviene per il software proprietario - solo in caso di diffusione commerciale. In questo caso è l’azienda che realizza la “distribuzione” che esegue le verifiche ed i test sul software utilizzato e che si assume la responsabilità di fornire una garanzia “limitata”.

La limitazione scelta nella maggior parte dei casi fa sì che si debba solo risarcire il prezzo del pacchetto software, oppure obbliga alla sostituzione del supporto danneggiato. Mai, in nessun caso, viene riconosciuto un risarcimento per il danno causato per la perdita di dati o altri danni causati da malfunzionamento del software. Ma questo è vero anche per i prodotti commerciali proprietari, quindi da questo punto di vista non si riscontrano significative differenze tra il software libero e quello proprietario.

- Software Open Source. Tale dizione è frutto di una rivisitazione delle proposte della Free Software Foundation operata nel 1998 da un gruppo di persone (fra cui Todd Anderson, Chris Peterson e Eric Raymond), con l’obiettivo di avvicinare lo sviluppo del software libero a quello sviluppato dalle imprese commerciali<sup>49</sup>, permettendo a queste ultime di poter accedere e utilizzare i prodotti open source per coglierne le opportunità, e quindi favorire la diffusione e l’uso di tali soluzioni. Questo ha portato all’elaborazione delle linee guida per la compilazione delle licenze, la cui prima versione risale al 1997, anno in cui Bruce Perens scrisse “The Debian Free Software Guidelines” (DFSG)<sup>50</sup>, che in sintesi, si articola in dieci punti:

- *libertà di redistribuzione;*
- *libertà di rendere disponibile il codice sorgente senza alcun aggravio;*

---

protezione giuridica nonostante l’obiettivo sia “inverso” rispetto a quello del software proprietario. Per ottenere il suo scopo, il Permesso d’autore, dopo le clausole espresse di consenso all’uso libero, cioè, di eseguire, copiare, modificare e redistribuire senza restrizioni appone un’altra clausola, che rende il copyleft così originale ed efficace, che vieta di aggiungere restrizioni ed obbliga ad ogni passaggio di trasmettere esattamente le stesse libertà che si erano ricevute. Il copyleft consente al software GNU di non essere “snaturato”, di mantenere, cioè, l’idea originaria di bene pubblico puro. Le libertà date dal Permesso d’autore, diventano, così, dei diritti inalienabili. Esempi di licenze copyleft per il software sono la GNU GPL e, in minor misura, la GNU LGPL, per altri ambiti le Creative Commons License con la clausola share alike). Il testo ufficiale della licenza è disponibile all’URL <http://www.freebsd.org/copyright/index.html> ed in particolare: <http://www.freebsd.org/copyright/license.html>.

<sup>49</sup> In realtà molte aziende private già investivano nella realizzazione/evoluzione di software libero, ma la connotazione “ideologica” della FSF scoraggiava alcune aziende dal manifestare il loro interesse per questo tipo di approccio.

<sup>50</sup> Maggiori dettagli sono disponibili sul sito ufficiale Debian all’URL [http://www.debian.org/social\\_contract.html](http://www.debian.org/social_contract.html)

- *possibilità di modifica o di creazione di opere derivate, con la garanzia di poterle distribuire con la licenza del software originario;*
- *salvaguardia dell'integrità del codice sorgente originario;*
- *nessuna discriminazione contro persone o gruppi;*
- *nessuna discriminazione degli ambiti o delle finalità di impiego;*
- *perpetuazione della licenza in modo che gli utenti finali non debbano accettare un nuovo accordo quando ricevono il software da un soggetto diverso dall'autore;*
- *assenza di prerequisiti fra programma e prodotti specifici, hardware e/o software;*
- *assenza di "contagio" delle licenze di altri programmi non open source sulla distribuzione di soluzioni open source;*
- *tipi di licenze da utilizzare come esempio (GPL, BSD).*

Quindi si può dire che il software regolato da licenza tipo GPL è open source, ma non è detto che qualunque licenza open source sia Free Software, né il viceversa<sup>51</sup>.

Si forniscono in sintesi alcune definizioni utili per la classificazione del software<sup>52</sup>:

- Public Domain, ovvero software che, per concessione dell'autore, non contiene alcuna indicazione di copyright, e che perciò è possibile utilizzare, riprodurre e diffondere liberamente, ma anche eventualmente modificare e inglobare all'interno di altro software non di public domain (anche commerciale);
- Freeware, cioè software con riserva di copyright per il quale non viene posto alcun particolare limite all'uso, la copia e la distribuzione. Nei casi in cui l'autore di freeware mette a disposizione anche il codice sorgente, in genere autorizza a modificarlo, ma non a utilizzarlo per scopi commerciali;
- Ad-ware, ovvero software gratuito che per l'utilizzo visualizza banner pubblicitari o installa nel computer dell'utente programmi spyware (software che raccolgono e inviano dati per usi solitamente commerciali);
- Demo, è il software generalmente di tipo commerciale che il fornitore ritiene utile distribuire in versione dimostrativa, con alcune funzioni inibite, o che cessa di funzionare dopo un periodo di tempo concesso per provarlo (questo secondo tipo è detto anche Shareware); ovvero, con alcune limitazioni d'uso (quali, ad esempio, il divieto di utilizzare il software in ambienti di produzione, ma solo in ambienti di

<sup>51</sup> Nessuna delle due organizzazioni ha un meccanismo di approvazione delle licenze completamente trasparente. La FSF non fa mistero del fatto che le licenze non sono valutate unicamente per come sono scritte, ma anche per le loro "intenzioni". Il Processo di valutazione della Open Source Iniziative (OSI - <http://www.opensource.org/index.php>) è formalizzato, ma ci sono alcuni passaggi che la procedura stessa descrive come "outbound", segno che anche il loro approccio non è del tutto cristallino. In definitiva, tenendo conto della vaghezza delle definizioni da una parte, e dei meccanismi di valutazione dall'altra, le uniche considerazioni ineccepibili circa l'intersezione/inclusione sono quelle a posteriori.

<sup>52</sup> Per una buona classificazione dei tipi di software, si rimanda alle definizioni (in italiano) che si trovano all'URL <http://www.gnu.org/philosophy/categories.it.html>. Queste sono, generalmente, accettate e riconosciute.

laboratorio, progetti prototipali o progetti pilota), rispetto alla versione commerciale (piena) che può essere utilizzata in produzione. Dipende, quindi, da caso a caso se questo tipo di software può essere usato, copiato e distribuito liberamente, o no, concordemente con le clausole contrattuali siglate dal cliente con il fornitore;

- Shareware, sono i pacchetti software che si possono ottenere gratuitamente per essere provati per un certo periodo di tempo. Allo scadere del periodo di prova, occorre pagare una certa somma all'autore, altrimenti il programma cessa di funzionare;
- Commerciale, è il software rilasciato da una azienda per essere venduto su licenza, con o senza periodi di prova. Prevede il pagamento per l'uso della licenza da parte dell'utente per poter entrare in possesso del supporto e/o per poter cominciare l'utilizzo del pacchetto software. L'accettazione della licenza e delle clausole contrattuali da parte dell'utente avviene, alcune volte, per il solo fatto di aprire la confezione esterna in plastica trasparente del package, che equivale alla lettura e accettazione delle clausole contrattuali leggibili all'esterno della scatola, ovvero, con l'accettazione esplicita delle clausole contrattuali durante l'installazione del prodotto. Il software commerciale non consente la distribuzione libera ad altri utenti, anche se interni alla stessa organizzazione, ma è possibile distribuire, o consentire l'uso del software ad altri utenti solo se vengono acquistate le relative licenze;
- Software “libero in parte”. Esistono alcune licenze che consentono un utilizzo libero del software solo in particolari condizioni, o all'interno di particolari gruppi di utenti (ad esempio, all'interno di università e centri di ricerca);
- Free for personal use, è l'estensione del concetto di uso personale. Dipende dalla licenza d'uso particolare. In linea di massima si può intendere che sia consentito l'utilizzo del pacchetto software al di fuori dell'attività lavorativa, ovvero, un uso non per scopi professionali (ad esempio, molti “compilatori” vengono offerti con licenza “free for personal use” per consentire a studenti e persone che operano a livello hobbistico di poter sviluppare dei programmi e compilarli sul proprio computer. Gli stessi compilatori ed ambienti di sviluppo sono venduti, su licenza, alle aziende e professionisti che operano nel campo dello sviluppo del software).

### **1.7. Il modello open source e il mercato**

Per i fornitori, l'utilizzo esclusivo di applicazioni open source implica, di fatto, la rinuncia ad avvalersi della protezione del software garantita “implicitamente” dal segreto industriale della codifica (in quanto si distribuisce il solo oggetto eseguibile).

L'azienda produttrice, che decide di distribuire anche il codice sorgente, se da una parte cooperando con altre aziende/soggetti può avvantaggiarsi della collaborazione anche di terzi, dall'altra accetta di spostare la competizione commerciale dal vantaggio derivante dall'essere l'unico soggetto che detiene diritti esclusivi sul proprio codice sorgente<sup>53</sup> a quello della “qualità dei servizi” offerti alla clientela (rispetto a quelli offerti dagli altri operatori di mercato).

<sup>53</sup> La conoscenza “esclusiva” del codice sorgente è solo uno dei vantaggi. Un ulteriore vantaggio, forse quello più strategico è quello di poter modificare il sorgente senza dover chiedere il permesso ad altri soggetti. Questo può essere utilizzato come fattore competitivo ma potrebbe essere utilizzato per spiazzare i concorrenti o altre aziende che dipendono da quel software.

Ciò non implica, in ogni caso, che le aziende informatiche debbano decidere necessariamente solo tra due possibili alternative: software proprietario o software libero. È, infatti, possibile seguire percorsi commerciali che prevedono l'uso di entrambe le categorie di software (proprietario e libero), che consentono di tutelare ciò che si pensa di poter mantenere come “vantaggio competitivo” e pertanto si ritiene di non poter diffondere.

Nel caso in cui l'azienda decida di partecipare, ovvero di far aderire, la comunità di sviluppatori a un proprio progetto, diffonderà (anche protetto dalla legge del Diritto d'Autore) il listato della codifica che intende condividere. Ciò le consentirà di beneficiare del *modello cooperativo di sviluppo* tipico della comunità OS. Infatti, in ambito OS si ha cessione gratuita perché la modalità di allocazione delle risorse è la reciprocità: la comunità dona il software all'utente e l'utente fa aumentare il valore intrinseco del software in quanto ne aumenta la diffusione. In più, se l'utilizzatore è un utente attivo, egli arricchisce il patrimonio di conoscenza, il capitale sociale che sta dietro il progetto di sviluppo di quel software, e arricchisce quindi il progetto stesso nel suo complesso.

Quindi, scegliere di distribuire software con licenze di tipo OS può avere anche un'utilità commerciale, in quanto il diffondersi di questo tipo di distribuzione del software favorisce l'esternalità di rete di quei produttori che hanno liberalizzato le proprie soluzioni, soprattutto se sono stati i primi a rilasciare quel tipo di programma.

La distribuzione gratuita del software è, pertanto, una strategia attuata sia in ambito proprietario, sia in ambito OS. Ciò che le distingue è la finalità sottostante. In ambito proprietario si mira alla massima diffusione per conquistare fette di mercato, legando l'utenza ai propri prodotti e favorendo l'affermazione dei propri standard, per “abituare” gli utenti all'utilizzo di certi tipi di soluzioni, essendo consci che migrare da un software che si “conosce” a un altro, anche se svolge lo stesso compito, è costoso quantomeno per il tempo da dedicare all'apprendimento da parte degli utenti. E ancora, si mira alla massima diffusione per facilitare la commercializzazione di altro software o la versione più evoluta (o completa) di altro software prodotto dall'azienda.

Il software open source impone alle aziende informatiche di trovare un modo diverso di “fare business” con i propri clienti.

Il modello OS consente alle aziende che sviluppano software di spostare il loro “core business” dalla produzione di codice sorgente, che può essere estremamente onerosa e ad alto rischio per l'impresa (in quanto non sempre il prodotto ottiene il riconoscimento dal mercato necessario a remunerare l'investimento), verso attività di fornitura servizi, di integrazione di soluzioni software e di edizioni personalizzate per l'utente di questo software.

La produzione interna di codice non cessa, ma viene parzialmente esternalizzata e reindirizzata verso le comunità di sviluppo, sotto forma di contributo collaborativo volontario. Il contributo non è “a fondo perduto” perché i propri tecnici, collaborando, aumentano le proprie competenze e conoscenze che poi riportano all'interno dell'azienda come patrimonio di conoscenza.

L'elemento importante nel modello OS, che molti riconoscono come premiante, è quindi quello della valorizzazione del supporto tecnico e delle attività di system integration. Si tratta, in altre parole, di portare sul mercato, sotto forma di business, non il valore dell'acquisto del programma e del sorgente, ma quello di servizi di supporto informatico di qualità, per soddisfare le specifiche esigenze dell'utente.

Ulteriore aspetto dell'uso di software open source riguarda la “neutralità tecnologica” (non proprietaria) che consente di condividere i formati e gli standard aperti. Questo favorisce, indirettamente, la standardizzazione delle soluzioni informatiche, e consente di poter scegliere quelle tecnologie che non inibiscono la comunicazione tra entità che utilizzano piattaforme diverse.

Infine, si evidenzia il problema della “protezione degli investimenti” per gli utenti. In ambiente proprietario si è legati indissolubilmente al fornitore del software, quindi si è costretti a cambiare il prodotto se questo non è più supportato dal fornitore. Altro problema nasce se il fornitore cessa la sua attività, o cede ad altra azienda le attività, e quindi l'utente si trova a dover affrontare una migrazione verso un'altra soluzione, ovvero a dover operare con un fornitore diverso senza avere scelto di effettuare il cambiamento.

### 1.7.1.1 “prodotti” open source

Le “distribuzioni”<sup>54</sup> sono l'esempio più significativo di attività commerciale da parte di organizzazioni che si dedicano al software open source. La distribuzione si occupa di assemblare le varie parti del sistema (kernel e software di supporto), di sviluppare quello che manca o che si vuole migliorare, di mettere insieme una collezione di software applicativo e di verificare che il tutto funzioni senza problemi.

Generalmente le distribuzioni (Distro) differiscono tra loro per varie caratteristiche, come ad esempio i metodi di installazione, gli strumenti per l'amministrazione del sistema ed altro. Ognuna è orientata ad una o più nicchie di mercato, per cui si hanno distribuzioni dedicate alle aziende<sup>55</sup>, al desktop<sup>56</sup> e altro<sup>57</sup> ancora.

Le distribuzioni<sup>58</sup> partecipano attivamente ai progetti OS collegati al loro prodotto, o sono esse stesse dei progetti.

Le organizzazioni commerciali, o le aziende, che seguono questo modello, devono offrire alla propria clientela un valore aggiunto. Non si possono appoggiare alla semplice cessione della licenza d'uso come nel modello proprietario. Molte riescono a ottenere buoni utili da fonti quali:

- la vendita di copie;
- la vendita di manualistica e materiale di supporto;
- lo sfruttamento del marchio;
- la garanzia;
- la formazione;
- i servizi di consulenze, sviluppo e supporto tecnico post-vendita.

---

<sup>54</sup> Con il termine “distribuzione” si indica sia il prodotto che l'azienda che lo rilascia.

<sup>55</sup> Ad esempio: Red Hat (che è anche la più diffusa tra le commerciali), S.u.S.E., ecc.

<sup>56</sup> Ad esempio: Mandrake, Debian, ecc.

<sup>57</sup> Ad esempio: i progetti avviati da Debian dedicati ai bambini tra i 7 e i 12 anni (Debian Jr.), alla pratica medica e di ricerca (Debian Med), al settore educativo (Debian-Edu), ecc.

<sup>58</sup> Si evidenzia, per distinguerla dalle altre, la distribuzione Debian GNU/Linux, che esiste e vive solo sui canali di comunicazione (Internet). È un sistema operativo completamente libero, sviluppato in modo cooperativo da un migliaio circa di volontari sparsi in tutto il mondo, che collaborano via Internet.

Le versioni “shrink wrap” vendute nei negozi includono copie del sistema e degli applicativi, manualistica e servizi come l’assistenza (telefonica, on-line, oppure on-site) o la garanzia, il supporto tecnico ed eventualmente software proprietario di cui rivendono le licenze.

La vendita si compone di due categorie generali di prodotto: beni fisici e servizi. Le versioni liberamente scaricabili dalla rete sono gratuite ma non includono servizi extra.

#### **1.7.2.1 “prodotti accessori”**

Intorno al modello OS sono sorte numerose imprese individuali, o aziende, che dedicano la propria attività, ricavando profitto, dalla fornitura di servizi (supporto, ricerca, sviluppo, localizzazione, personalizzazione, formazione, ecc.), oppure fornendo manuali sul software OS, pubblicati con licenze che consentono la riproduzione dei testi come la GNU/GPL o la GNU/FDL.

#### **1.7.3.1 consulenti autonomi**

La disponibilità dei sorgenti e la modularità, che rendono estremamente flessibili il software OS, consentono di adattare (personalizzare) le funzionalità della soluzione software per le diverse necessità dei singoli utenti. Quindi, l’utente che ha particolari esigenze può rivolgersi a consulenti autonomi in grado di fornire assistenza tecnica per personalizzare ed eventualmente estendere le funzionalità del software con le caratteristiche di cui ha bisogno.

È una versatilità diversa rispetto al software proprietario, che può essere anch’esso personalizzato, ma solo per le funzionalità concesse (previste) dal produttore. Inoltre, l’assistenza al software proprietario è spesso fornita soltanto da consulenti autorizzati (certificati e formati sulla specifica piattaforma).

Il software OS, invece, consente libertà di scelta tra consulenti, purché questi abbiano la conoscenza tecnica necessaria per operare sul software. L’offerta di consulenti a supporto del software OS può adattarsi alla domanda del mercato e le competenze disponibili sul mercato possono evolvere parallelamente e in conseguenza all’evoluzione del software.

#### **1.7.4. Le società di assistenza tecnica**

Qualora il bisogno del cliente non possa essere coperto da una singola persona o da una piccola impresa, l’utente si può rivolgere a una società specializzata che offra servizi di system integration e assistenza tecnica.

Come accennato prima, in situazioni di questo tipo l’uso di software proprietario vincola l’utente al servizio offerto dal fornitore, o da aziende da questo autorizzate, limitando la possibilità di scelta. L’utilizzazione di software OS consente una più ampia scelta del fornitore del servizio, che può essere lo stesso che ha rilasciato il software, un suo business partner, oppure una società indipendente che opera sul mercato, con pari opportunità con tutte le altre imprese di servizi.

La diffusione di software OS può svolgere, quindi, un ruolo importante per la rivalutazione dei servizi professionali legati al mondo dello sviluppo del software, e può rappresentare un’importante opportunità di crescita per un effettivo mercato locale di integratori di sistema, ISV (independent software vendor) e software house.

Altro aspetto tecnico, relativo all'adozione di soluzioni OS, riguarda la maggiore flessibilità, per gli ISV e le software house, di poter realizzare le proprie soluzioni senza dover sottostare, ad esempio, a clausole contrattuali e restrizioni imposte da fornitori di software proprietario. Queste aziende potranno scegliere piattaforme informatiche che non implicano l'acquisto di licenze d'uso, potranno utilizzare della piattaforma prescelta solo le funzionalità loro necessarie, a beneficio della soluzione applicativa da loro sviluppata. Questa potrà essere proposta agli utenti senza imporre, ad esempio, l'acquisto di specifiche piattaforme.

### **1.7.5. Le società di formazione e certificazione**

Si indicano con tale nome le società indipendenti che formano i tecnici e attestano la loro preparazione. Sono già presenti sul mercato numerosi operatori che offrono questo tipo di servizio.

Per quanto riguarda, in generale, il software OS, sono presenti sul mercato numerose società in grado di fornire attestati e garanzie di funzionamento, compatibilità, standardizzazione.

### **1.7.6. Le “garanzie” sul software**

Si evidenzia che con l'acquisto di una licenza software non si ottiene automaticamente la “certezza” di ricevere risposte ai propri bisogni. Nessun fornitore garantisce che il proprio prodotto software assolverà ad alcuna funzione, compresa quella per cui lo si è acquistato. È un'immunità comprensibile per un libro, un quadro o per una canzone, ma non per un bene strumentale come il software.

Legalmente non si può chiamare in giudizio il produttore del software, nemmeno se si scoprono palesi errori di programmazione<sup>59</sup> o, peggio, se si è subito un danno (come la perdita di dati) causato dal suo provato malfunzionamento. Conseguenza diretta di questo stato di cose è che il produttore di software non è tenuto (obbligato da alcuna clausola contrattuale) a correggere errori segnalati, o riconosciuti.

Non tutti i produttori possono avvantaggiarsi di queste specificità giuridiche, ma solo quelli con grande potere contrattuale; il piccolo operatore di mercato sarà comunque

---

<sup>59</sup> Il D.P.C.M. 06/08/1997 n. 452, *Regolamento recante approvazione del capitolato di cui all'articolo 12, comma 1, del D.Lgs. 12 febbraio 1993, n. 39, relativo alla locazione e all'acquisto di apparecchiature informatiche, nonché alla licenza d'uso dei programmi*. G.U. 30 dicembre 1997, n. 302. Testo 2. Specificamente il Capo III.

È vero, infatti, che il teorema di Rice afferma che per un generico programma non è semplice stabilire se sia o meno coerente con le sue specifiche (a meno che esse stesse non siano un programma, ndr), dal punto di vista puramente teorico la correttezza e la completezza di un programma in generale non sono “dimostrabili”. Per questa ragione il software deve essere testato e collaudato dall'amministrazione, come stabilito dall'Articolo 36: “Prova e collaudo dei programmi” – dopo di che la stessa li accetta, ovvero, se non soddisfano li deve restituire come stabilito al par.3 - *L'amministrazione restituisce i programmi non giudicati adatti alle proprie esigenze non oltre quindici giorni dalla scadenza del periodo di prova, riconoscendo all'impresa, per tali programmi, solo il corrispettivo dell'installazione, nel caso in cui questa sia avvenuta a sua cura e spese*.

Inoltre, l'articolo 38: “Modifiche ai programmi” - è in parte inapplicato nella prassi, soprattutto per il par.4 - *Nel caso in cui l'amministrazione ritenga di continuare ad operare con la versione originaria dei programmi, deve darne comunicazione all'impresa, la quale rimane impegnata a continuare a prestare, su tale versione, i servizi e la collaborazione di supporto in atto*, infatti, quando un prodotto non è più supportato l'amministrazione è di fatto costretta ad effettuare una migrazione ad una nuova versione o ad un nuovo prodotto.

Per quanto riguarda invece gli aspetti non tecnici, le licenze proprietarie escludono esplicitamente ogni responsabilità, o come in alcuni casi ne accettano entro determinati limiti (es.: xxx dollari), ma il fatto che non ci sia una ricca giurisprudenza in merito rende eventuali contestazioni legali di difficile risoluzione.



tenuto ad accollarsi i costi relativi all'obbligo di risultato e alle clausole di garanzia per poter lavorare e per soddisfare le commesse importanti dei suoi clienti.

Allo stesso modo, l'utente "medio" ha poteri contrattuali molto limitati. Ad esempio, quando le software house ritirano dal commercio i vecchi programmi, non hanno (a meno di accordi/clausole specifiche) l'obbligo di depositare i sorgenti e/o le specifiche tecniche in archivi pubblici. In tal caso, nessuno più potrà supportare gli utenti di quei programmi.

Con l'adozione di soluzioni OS, il problema di fatto è immutato. L'unica differenza consiste nel fatto che, essendo il codice sorgente accessibile, è possibile in teoria individuare sul mercato un esperto in grado di risolvere quel problema. Ma questo è un concetto di principio e non costituisce comunque una garanzia.

## **2. Metodologia di attuazione della Direttiva del MIT**

### **2.1. La Direttiva del Ministro Stanca**

Il 19 dicembre 2003 il Ministro per l'Innovazione e le Tecnologie ha emanato una direttiva sullo sviluppo e l'utilizzazione dei programmi informatici da parte delle pubbliche amministrazioni. Tale direttiva (da questo punto in avanti, per comodità, ci si riferirà ad essa come "Direttiva") è stata pubblicata sulla Gazzetta Ufficiale n. 31 del 7/2/2004.

La Direttiva fornisce alle pubbliche amministrazioni indicazioni nonché criteri tecnici e operativi per gestire più efficacemente il processo di predisposizione o di acquisizione di programmi informatici. In particolare, essa indica come le pubbliche amministrazioni debbano tener conto della offerta sul mercato della modalità di sviluppo e diffusione di programmi informatici definita "open source".

Per entrare più in dettaglio sull'argomento, si riportano nel seguito alcuni passaggi del testo della Direttiva, in particolare gli articoli 3, 4 e 7.

#### ***Art 3. Analisi comparativa delle soluzioni.***

*1. Le pubbliche amministrazioni, nel rispetto della legge 7 agosto 1990, n. 241 e del decreto legislativo 12 febbraio 1993, n. 39, acquisiscono programmi informatici a seguito di una valutazione comparativa tra le diverse soluzioni disponibili sul mercato.*

*2. In particolare, valutano la rispondenza alle proprie esigenze di ciascuna delle seguenti soluzioni tecniche:*

*a) sviluppo di programmi informatici ad hoc, sulla scorta dei requisiti indicati dalla stessa amministrazione committente;*

*b) riuso di programmi informatici sviluppati ad hoc per altre amministrazioni;*

*c) acquisizione di programmi informatici di tipo proprietario mediante ricorso a licenza d'uso;*

*d) acquisizione di programmi informatici a codice sorgente aperto;*

*e) acquisizione mediante combinazione delle modalità di cui alle lettere precedenti.*

*3. Le pubbliche amministrazioni valutano quale soluzione, tra le disponibili, risulta più adeguata alle proprie esigenze mediante comparazioni di tipo tecnico ed economico, tenendo conto anche del costo totale di possesso delle singole soluzioni e del costo di uscita. In sede di scelta della migliore soluzione si tiene altresì conto del potenziale interesse di altre amministrazioni al riuso dei programmi informatici, dalla valorizzazione delle competenze tecniche acquisite, della più agevole interoperabilità. La prospettazione degli elementi di cui sopra è peraltro oggetto di valutazione da parte del Centro nazionale per l'informatica nella pubblica amministrazione in sede di rilascio del parere di cui all'art. 8 del decreto legislativo 12 febbraio 1993, n. 39. La suindicata valutazione va inclusa nell'ambito dello studio di fattibilità prescritto dall'art. 13 del decreto legislativo 12 febbraio 1993, n. 39, allorché si tratti di contratti di grande rilievo.*

#### **Art 4. Criteri tecnici di comparazione.**

*Le pubbliche amministrazioni, nella predisposizione o nell'acquisizione dei programmi informatici, privilegiano le soluzioni che presentino le seguenti caratteristiche:*

- a) soluzioni informatiche che, basandosi su formati dei dati e interfacce aperte e standard, assicurino l'interoperabilità e la cooperazione applicativa tra i diversi sistemi informatici della pubblica amministrazione, salvo che ricorrano peculiari ed eccezionali esigenze di sicurezza e segreto;*
- b) soluzioni informatiche che, in assenza di specifiche ragioni contrarie, rendano i sistemi informatici non dipendenti da un unico fornitore o da un'unica tecnologia proprietaria; la dipendenza è valutata tenendo conto dell'intera soluzione;*
- c) soluzioni informatiche che, con il preventivo assenso del C.N.I.P.A. e in assenza di specifiche ragioni contrarie, garantiscano la disponibilità del codice sorgente per ispezione e tracciabilità da parte delle pubbliche amministrazioni, ferma la non modificabilità del codice, fatti salvi i diritti di proprietà intellettuale del fornitore e fermo l'obbligo dell'amministrazione di garantire segretezza o riservatezza;*
- d) programmi informatici che esportino dati e documenti in più formati, di cui almeno uno di tipo aperto.*

#### **Art 7. Riuso.**

*1. Al fine di favorire il riuso dei programmi informatici di proprietà delle amministrazioni, nei capitolati o nelle specifiche di progetto dovrà essere previsto, ove possibile, che i programmi sviluppati ad hoc siano facilmente portabili su altre piattaforme.*

### **2.2.Impatto della Direttiva sulle procedure di acquisizione**

Con riferimento agli articoli riportati nel paragrafo precedente, il primo aspetto da considerare è che la Direttiva non intende stravolgere le usuali procedure di acquisizione di programmi informatici da parte delle amministrazioni pubbliche, come rappresentato dal riferimento alle leggi vigenti 241/1990 e 39/1993.

Esaminando l'articolo 3, la novità più significativa sembra rappresentata dall'elenco di soluzioni tecniche possibili, che includono (per la prima volta esplicitamente) il riuso di applicazioni sviluppate per altre amministrazioni e programmi open source. Secondo quanto espresso dalla Direttiva, le amministrazioni che intendano dotarsi di un sistema informatico dovranno effettuare (e riportare nello Studio di Fattibilità) una comparazione tra le soluzioni disponibili, sulla base anche dei seguenti elementi:

- TCO (costo totale di possesso) della singola soluzione;
- costo di uscita dalla stessa;
- valorizzazione delle competenze tecniche possedute dall'amministrazione;
- interoperabilità intesa per la PA nel suo complesso (dunque uso di formati dei dati e interfacce aperte e standard);

- interesse di altre amministrazioni al riuso dell'applicazione da realizzare e/o acquistare.

La Direttiva non entra nel merito di come eseguire la comparazione di cui sopra: afferma soltanto che la stessa sarà oggetto di valutazione da parte del CNIPA in sede di rilascio del Parere di congruità tecnico/economica. Ciò comporta che, a oggi:

- le amministrazioni non hanno una chiara indicazione su come effettuare la comparazione;
- il CNIPA non dispone di riferimenti con cui valutare la comparazione effettuata.

Per questi motivi sembra necessario definire criteri univoci e proporre una metodologia cui attenersi per compiere la comparazione. In tal modo:

- le amministrazioni disporranno di linee guida da seguire, facoltativamente, nell'ambito della loro autonomia (che la Direttiva non intende porre in discussione),
- il CNIPA potrà valutare i risultati della comparazione effettuata dall'amministrazione anche tenendo conto degli eventuali scostamenti, motivati o meno da parte dell'amministrazione, rispetto alla metodologia proposta.

L'articolo 4 della Direttiva fornisce una serie di caratteristiche tecniche sulla base delle quali motivare la scelta di soluzioni informatiche da parte delle amministrazioni.

Nessuna di tali caratteristiche è qualificata dalla Direttiva come obbligatoria. Tuttavia la loro presenza viene indicata come criterio di preferenza: a parità di altre condizioni, è favorita la soluzione che fornisce il maggior numero delle caratteristiche indicate.

In sintesi, le caratteristiche tecniche indicate nella Direttiva sono:

- uso di standard aperti per dati e interfacce;
- indipendenza da tecnologie proprietarie o dipendenti da un unico fornitore;
- disponibilità del codice sorgente per ispezione e tracciabilità<sup>60</sup>;
- capacità di esportare dati e documenti in formato aperto.

Si noti che le caratteristiche tecniche di cui sopra non riguardano solo i prodotti da acquisire, ma sono valide anche per le realizzazioni di soluzioni informatiche ad hoc: diverranno in questo caso requisiti che il fornitore (nel caso di sviluppo esterno) dovrà garantire per contratto.

---

<sup>60</sup> questo punto, come segnalato da alcuni commentatori, si presta a interpretazioni non allineate all'effettivo significato. Poiché la disponibilità del codice sorgente è insita nella natura stessa dei software open source (in caso contrario non potrebbero essere definiti tali), per questo tipo di prodotti il punto in questione è privo di significato, mentre ha senso e si applica ai software proprietari. In pratica significa che, tra i vari prodotti software proprietari, sono da preferire quelli che garantiscono la disponibilità del codice sorgente. Restando comunque software di tipo proprietario, tali prodotti garantiranno l'ispezione e la tracciabilità del sorgente, ma non la sua modificabilità; l'amministrazione dovrà garantire segretezza e riservatezza. Ciò potrà avvenire tramite accordi tra l'amministrazione, il fornitore e un eventuale soggetto terzo garante (il CNIPA)

Si noti inoltre che nessuna delle caratteristiche di cui sopra esclude a priori un particolare fornitore o una particolare tipologia di soluzione informatica. Sono comunque previste eccezioni (specifiche ragioni contrarie) per cui un'amministrazione può scegliere, motivando la decisione, di acquisire una soluzione informatica che non disponga delle caratteristiche indicate. Sarà poi il CNIPA, in sede di Parere, a valutare le motivazioni espresse in tali casi dall'amministrazione.

### **2.2.1. Studio di fattibilità**

Nel corso del processo di acquisizione e/o realizzazione di una soluzione informatica, il primo passo su cui la Direttiva ha impatto è la stesura dello Studio di Fattibilità.

Lo Studio di Fattibilità è un documento di progetto reso obbligatorio per la prima volta, e solo per i contratti di grande rilievo, dall'articolo 13 del decreto legislativo 12 febbraio 1993, n. 39.

Alcuni anni dopo, in ultima edizione nel marzo del 1997, l'AIPA (oggi CNIPA) pubblicò un insieme di linee guida per la realizzazione di Studi di Fattibilità relativi a progetti di adeguamento dei sistemi informativi automatizzati, e in particolare a progetti per la realizzazione di sistemi informatici. Tali linee guida sono indirizzate ad amministrazioni centrali dello Stato e degli Enti pubblici non economici nazionali, individuati nell'articolo 1 del decreto legislativo 12 febbraio 1993, n. 39.

Le linee guida di cui sopra propongono una struttura (indice tipo) per gli Studi di Fattibilità, all'interno della quale un punto essenziale è l'esame delle alternative di progetto. Si riporta un estratto di tale passaggio, allo scopo di esaminarlo e individuare affermazioni e concetti da aggiornare, modificare o estendere alla luce del contenuto della Direttiva.

*Le alternative da considerare debbono essere situate all'interno dei requisiti definiti, ossia debbono essere alternative comunque "efficaci", altrimenti è inutile valutarle.*

*Una volta esaminate e valutate le alternative, la scelta operata diventa requisito vincolante del progetto. È da considerarsi eccezionale una valutazione di alternative diverse che porti solo alla indicazione di una generica preferenza, magari da utilizzare come elemento di valutazione delle offerte.*

*L'esame delle alternative si deve concentrare solo su aspetti effettivamente essenziali e determinanti la natura della soluzione individuata.*

...

*Lo studio di fattibilità dovrà esaminare e valutare compiutamente le alternative tecnologiche, attraverso una comparazione basata sul rapporto costi-benefici.*

*Esiste poi una ovvia possibilità di alternativa in termini di modalità e specifiche realizzative.*

*In questa area è necessario fare una distinzione tra scelte di tipo strategico, che modificano in maniera significativa la natura della soluzione che si sta definendo, e scelte che non incidono in maniera sostanziale nella definizione della soluzione, ma rappresentano soltanto diversi modi di realizzare la medesima soluzione.*

*Rientrano tipicamente nel primo gruppo ad esempio sia le scelte relative al “MAKE OR BUY”, sia le scelte sull’opportunità di recuperare o meno componenti del sistema esistente, magari con operazioni di incapsulamento o reingegnerizzazione.*

*Fanno parte tendenzialmente del secondo gruppo le scelte relative a strumenti e ambienti di*

*sviluppo, le scelte relative all’interfaccia utente, le scelte relative alle modalità di interconnessione ecc.*

*Lo studio di fattibilità si deve concentrare solo sulle alternative di tipo strategico, quelle che,*

*modificando la natura della soluzione, incidono su costi, tempi, caratteristiche essenziali di qualità del sistema che si vuole realizzare nonché, indirettamente, anche sui benefici.*

*Le varie modalità e specifiche realizzative, quando non incidono sostanzialmente sulla natura della soluzione stessa, non debbono quindi essere trattate come alternative, ma demandate sostanzialmente all’offerta dei possibili fornitori.*

*È infatti importante che si dia la possibilità ai fornitori di esprimere compiutamente nelle offerte la propria capacità progettuale, senza vincolarle in limiti angusti e non necessari. La proposizione dei fornitori, ovviamente principalmente in caso di progetti complessi e di forniture composite, rappresenta un patrimonio di conoscenza e di approfondimento a cui non si deve rinunciare in caso di gara, dato che può portare alla proposta di soluzioni che aggiungono qualità al sistema che si vuole realizzare.*

...

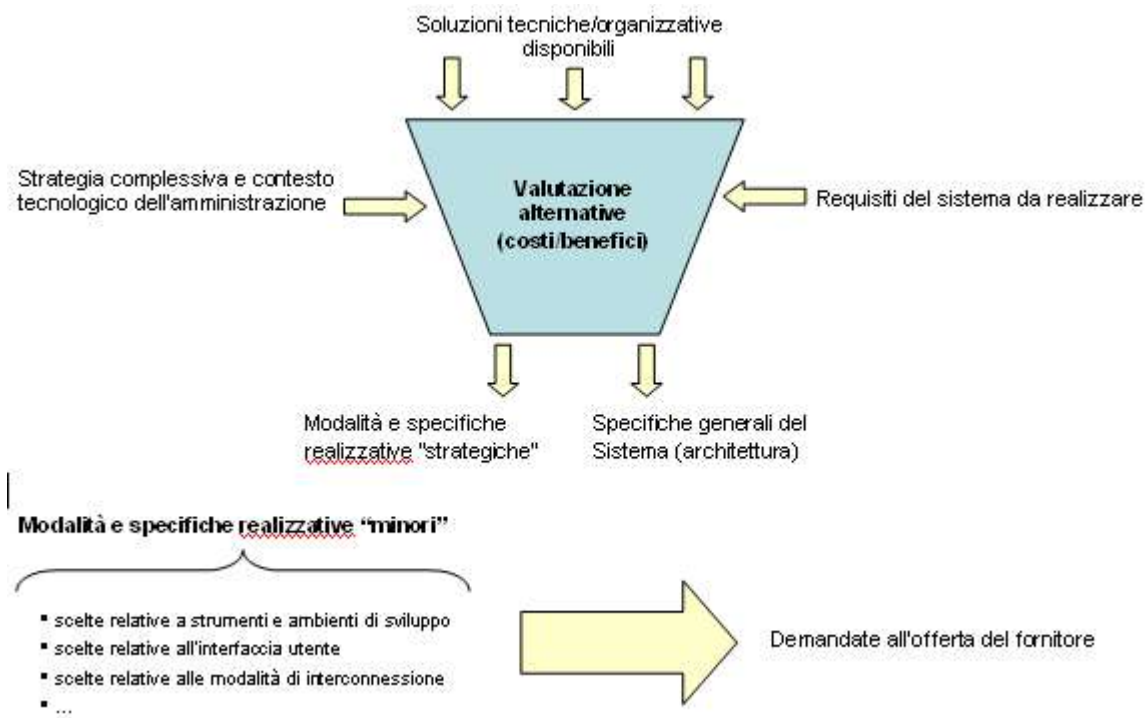
*La valutazione dovrà tener conto sia degli aspetti funzionali che degli aspetti tecnici, oltre naturalmente all’aspetto economico.*

*Dal punto di vista funzionale dovrà essere fatta una mappatura delle funzioni offerte dalle soluzioni disponibili con i requisiti, mentre dal punto di vista tecnico si dovrà mappare la coerenza dell’ambiente tecnologico previsto dalla soluzione con il contesto tecnico dell’amministrazione.*

...

*È evidente peraltro che la comparazione economica non riguarderà l’insieme del progetto ma soltanto la componente su cui si è individuata la possibilità di ipotesi differenti e le componenti di costo da essa influenzate.*

*La valutazione delle alternative “classica” segue perciò lo schema illustrato in figura 2.1 (ove viene tradotto in forma visiva quanto espresso nei passaggi appena riportati).*



**Figura 2.1: schema "classico" per la valutazione delle alternative**

Tale schema deve essere aggiornato per consentire l'applicazione delle indicazioni della Direttiva<sup>61</sup>.

Essenzialmente, si dovrà porre attenzione a che l'analisi costi/benefici si basi, per ciò che riguarda i costi, su:

- il TCO della soluzione esaminata, definito come l'insieme dei costi che nel corso dell'intera vita operativa della soluzione sarà necessario sostenere affinché essa sia utilizzabile proficuamente dall'utenza;
- il costo di uscita dalla stessa, inteso come l'insieme dei costi da sostenere per abbandonare una tecnologia o migrare verso una tecnologia o soluzione informatica differente (costi di conversione dati, di aggiornamento dell'hardware, di realizzazione interfacce, di formazione);

e comprenda, tra i benefici da prendere in considerazione:

- la valorizzazione delle competenze tecniche possedute dall'amministrazione;
- la possibilità che altre amministrazioni riutilizzino l'applicazione da realizzare e/o acquistare;
- l'interoperabilità per la PA nel suo complesso, intesa come capacità di sistemi informativi anche eterogenei di condividere, scambiare e utilizzare gli stessi dati e funzioni d'interfaccia, anche attraverso l'uso di formati di dati aperti (ovvero

<sup>61</sup> Si noti che l'ambito di applicazione della Direttiva si estende anche ai progetti non di grande rilievo, dunque anche ai casi in cui lo Studio di Fattibilità non è obbligatorio. Allo stesso modo, le indicazioni che vengono date nel seguito sono in generale applicabili anche ai progetti di acquisizione o di realizzazione di un sistema informatico che non vengono preceduti da uno Studio di Fattibilità formalizzato.

formati pubblici e documentati esaustivamente) e di standard (una specifica o norma condivisa, emanata da un ente predisposto o affermatosi di fatto).

Si noti che nello schema “classico” non vengono prese in considerazione le soluzioni basate su software open source e le soluzioni basate sul riuso applicativo, ma queste non vengono neppure escluse. Semplicemente le linee guida attuali non le citano perché:

- nel caso del software open source, il fenomeno era trascurabile all’epoca della stesura delle linee guida,
- nel caso del riuso informatico, mancava ancora uno strumento normativo che favorisse questo tipo di soluzioni (la norma che per la prima volta promuove il riuso informatico tra amministrazioni pubbliche è la 340/2000, articolo 25).

Per essere allineate alle indicazioni della Direttiva, le amministrazioni possono continuare a redigere lo Studio di Fattibilità secondo la struttura “classica”, facendo però attenzione a compiere l’analisi costi/benefici come sopra specificato, e a estendere la valutazione delle alternative alle due tipologie di soluzioni di cui sopra (riuso e open source).

In particolare, nella valutazione dell’ipotesi “riuso applicativo”, le amministrazioni devono prioritariamente verificare se è disponibile una soluzione informatica, tra quelle disponibili per il riuso, che possa adeguatamente rispondere ai requisiti del progetto. Le applicazioni riusabili individuate in questa ricerca devono essere esaminate per capire se possono inserirsi adeguatamente, e con convenienza economica, nel contesto tecnico e organizzativo dell’amministrazione. In tal caso, l’amministrazione si porrà nel ruolo di “soggetto ricevente” del meccanismo del riuso applicativo.

Se non viene individuata alcuna applicazione riusabile che sia adeguata e compatibile con i requisiti di progetto, l’amministrazione deve ancora valutare se porsi nel ruolo di “soggetto cedente” del meccanismo del riuso applicativo. L’amministrazione dovrà cioè stabilire se l’applicazione che si accinge a sviluppare (o a realizzare tramite personalizzazione di un pacchetto di mercato) è di interesse per il mercato del riuso. In tal caso l’amministrazione dovrà inserire, tra le specifiche del progetto, anche le caratteristiche tecniche e funzionali che rendono un software sviluppato ad hoc (o una personalizzazione di un pacchetto di base) facilmente riutilizzabile in altri contesti.

Nella valutazione dell’ipotesi “open source”, le amministrazioni possono considerare:

- l’utilizzo di singoli prodotti open source all’interno del sistema da realizzare,
- prendere in considerazione l’ipotesi di implementare l’intero sistema o parti di esso mediante un’attività di sviluppo di software ad hoc che utilizzi il modello di sviluppo open source (descritto al capitolo ).

Ciò comporta anche:

- la scelta tra le varie tipologie di licenza compatibili con il modello open source; l’amministrazione dovrà dunque valutare, sulla base delle caratteristiche del progetto, della propria strategia complessiva e di considerazioni giuridiche, se il software open source che intende sviluppare (o commissionare a terzi) debba essere rilasciato ad esempio con licenza GPL, BSD o altre licenze tra quelle descritte al capitolo ;



- la scelta se aderire attivamente a una comunità di sviluppatori del prodotto (o dei prodotti) open source in esame, se limitarsi a usufruire dei servizi offerti dalle suddette comunità, oppure se fondare una propria comunità (ad esempio per supportare gli sviluppi di un software ad hoc che si pensa di sviluppare secondo il modello open source).

Si noti che normalmente l'output dell'attività "valutazione delle alternative" di figura 2.1 sono solo le specifiche generali del sistema e le modalità realizzative strategiche. Non è previsto, dunque, che a questo livello si scelgano i singoli componenti del sistema: verrà definita la sua architettura di massima, e il resto sarà in genere demandato alla formulazione dell'offerta del fornitore.

Dunque le soluzioni che vengono messe a confronto nello Studio di Fattibilità non sono, nella normalità dei casi, del tipo: prodotto XXX o prodotto YYYY, bensì soluzioni del tipo (ad esempio):

- a) Reengineering del sistema attualmente in funzione;
- b) Riutilizzo dell'applicazione sviluppata dall'amministrazione X;
- c1) Realizzazione di un nuovo sistema mediante sviluppo ad hoc commissionato all'esterno, con architettura basata su standard aperti, interfaccia di tipo web, possibilità di I/O con il mondo legacy, rilascio in licenza GPL del software sviluppato;
- c2) Realizzazione di un nuovo sistema mediante personalizzazione di un pacchetto di mercato (proprietario o open source) in grado di soddisfare i requisiti.

Esistono comunque eccezioni, costituite da progetti che per particolari caratteristiche e dimensioni richiedono scelte di dettaglio già in questa fase.

### **2.2.2.Capitolato di gara**

Nei progetti di automazione che prevedono il ricorso al meccanismo della gara pubblica, le amministrazioni redigono di norma, tra i documenti previsti dal meccanismo concorsuale, il Capitolato Tecnico (o Allegato Tecnico al Capitolato di Gara).

Tale documento riporta in genere le richieste da inoltrare ai fornitori, in termini di:

- requisiti funzionali, architetturali e di qualità della fornitura;
- specifiche globali della fornitura (es. specifiche tecniche dei prodotti, modalità di erogazione dei servizi professionali);
- segmentazione del progetto;
- riepilogo delle acquisizioni e realizzazioni previste (configurazione del progetto);
- piano di massima del progetto (piano dei rilasci, punti di controllo, scadenze temporali);
- forme di negoziazione delle varianti di progetto.

Nel Capitolato, inoltre, vengono in genere definiti i criteri di selezione delle offerte, ovvero i criteri di assegnazione dei punteggi alle varie offerte pervenute. Tali criteri possono essere:

- economici (ovvero basati sul prezzo offerto);
- basati sulla qualità della proposta tecnica (nonché del piano di lavoro, della qualità del processo, ecc.);
- basati sulla valutazione del fornitore (es. qualità delle risorse coinvolte);
- su una combinazione dei tre tipi di criteri precedenti.

Anche in questo caso, è opportuno che la classica struttura e i tradizionali contenuti del Capitolato Tecnico vengano aggiornati per recepire le indicazioni della Direttiva.

In primo luogo, l'apertura a prodotti di tipo open source implica che la formulazione delle specifiche tecniche nel Capitolato non debba essere tale da escludere senza motivo prodotti del suddetto tipo. Ad esempio, una specifica formulata in questo modo: *“la fornitura dovrà comprendere un numero XX di PC Windows con Word e Outlook”* è errata. Una corretta formulazione potrebbe essere per esempio *“la fornitura dovrà comprendere un numero XX di personal computer provvisti di sistema operativo, word processor e client di posta elettronica”*.

Allo stesso modo, una specifica del tipo *“processore Intel a 3GHz”* è errata: il Capitolato dovrà limitarsi alla esposizione di specifiche funzionali e prestazionali, lasciando ai fornitori la possibilità di indicare prodotti diversi.

L'indicazione puntuale di un determinato prodotto potrà essere presente all'interno del Capitolato solo dietro motivazione tecnica o strategica. Ad esempio, se un'amministrazione ha già in esercizio un certo sistema operativo o un determinato DBMS, e ha costruito su questi strumenti una notevole esperienza, può ritenere opportuno indicare come specifica del nuovo sistema da realizzare l'utilizzo di quel sistema operativo o di quel determinato DBMS. In tal caso, nello Studio di Fattibilità l'amministrazione avrà effettuato la valutazione (con la relativa analisi costi/benefici) se continuare a servirsi del sistema operativo e DBMS già posseduti o se migrare ad altre soluzioni.

Un secondo impatto della Direttiva sui Capitolati Tecnici riguarda i criteri di assegnazione del punteggio alle proposte dei fornitori. Le caratteristiche tecniche indicate dalla Direttiva dovranno far parte, a meno di ragioni contrarie che andranno espresse, dei criteri di assegnazione del punteggio.

Un possibile esempio di elencazione dei criteri di assegnazione punteggio è riportato nella tabella seguente, in cui sono evidenziati il sottoinsieme di criteri introdotto dalla Direttiva.

**Tabella 2.2: criteri di assegnazione del punteggio**

<b><i>Qualità della proposta tecnica</i></b>	<b><i>Peso</i></b>	<b><i>Metodo di misura</i></b>
Comprensione del problema	10	Presenza riepilogo obiettivi, attività..
<b>Utilizzo formati e standard aperti</b>	15	Dichiarazione fornitore
Scalabilità della soluzione proposta	5	Valutazione della commissione

<b>Disponibilità del sorgente per ispezioni e tracciabilità</b>	5	Dichiarazione fornitore
<b>Disponibilità del sorgente per modifiche ed estensioni</b>	5	Dichiarazione fornitore
Strumenti utilizzati	10	Valutazione della commissione <sup>62</sup>
<b>Indipendenza da tecnologie proprietarie o di unico fornitore</b>	10	Valutazione della commissione
Funzionalità soluzione proposta	10	Valutazione della commissione
Affidabilità soluzione proposta	10	Valutazione della commissione
Usabilità soluzione proposta	10	Valutazione della commissione
<b>Predisposizione per il riuso</b>	10	Valutazione della commissione
<b>TOTALE</b>	100	
<b>Qualità del Fornitore</b>	<b>Peso</b>	
Curriculum delle risorse coinvolte	20	Presenza totale o parziale
<b>Competenze specifiche nelle tecnologie proposte</b>	20	Valutazione della commissione <sup>63</sup>
Esperienze progettuali	10	Citazione esperienze
Conoscenza contesto	10	Citazione esperienze
Congruenza piano di lavoro	10	Valutazione della commissione
Rilascio finale anticipato	10	Gg anticipo rispetto 160 gg max
<b>TOTALE</b>	60	

I pesi riportati in tabella sono stati scelti a puro titolo di esempio. In realtà essi saranno scelti dall'amministrazione sulla base di proprie considerazioni motivate. Nel prossimo paragrafo si suggerisce un'apposita metodologia, che l'amministrazione potrà facoltativamente utilizzare a tale scopo.

Qualora l'amministrazione abbia valutato opportuno o almeno plausibile, nello Studio di Fattibilità, utilizzare nel progetto anche software open source, l'amministrazione dovrà inserire tale indicazione all'interno del Capitolato Tecnico, invitando i fornitori a tenerne conto nella formulazione delle loro offerte.

In tali casi, sembra opportuno inoltre che il Capitolato inviti i fornitori a specificare in dettaglio, nella descrizione delle rispettive offerte tecniche, quali moduli software costituenti l'offerta sono di tipo proprietario e quali sono invece di tipo open source, con l'eventuale indicazione del perché è stata preferita una tipologia di software piuttosto che un'altra. Tali informazioni saranno d'ausilio all'amministrazione in sede di valutazione comparativa delle offerte, come sarà descritto nel prossimo paragrafo.

Infine, qualora l'amministrazione abbia valutato opportuno, o almeno plausibile, l'ipotesi di implementare l'intero sistema o parti di esso mediante un'attività di sviluppo di software ad hoc che utilizzi il modello di sviluppo open source (descritto al capitolo ), l'amministrazione dovrà inserire tale indicazione all'interno del Capitolato Tecnico, invitando i fornitori a tenerne conto nella formulazione delle loro offerte. Le problematiche e le opportunità legate a tale ipotesi realizzativa verranno esaminate in dettaglio nel paragrafo 2.4.

<sup>62</sup> Nel caso vengano proposti prodotti open source, valutazione della maturità degli stessi (vedi prossimo paragrafo)

<sup>63</sup> Nel caso vengano proposti prodotti open source, la valutazione avverrà anche sulla base della partecipazione del fornitore alla community e all'evoluzione del prodotto, nonché sulla base degli investimenti dichiarati dal fornitore sugli sviluppi e sul supporto del prodotto

### 2.3. Metodologia di valutazione comparativa dei prodotti software

La metodologia illustrata nel seguito ha lo scopo di supportare la valutazione comparativa tra le varie alternative disponibili per la realizzazione di un sistema informatico, con riferimento a:

- la determinazione dei pesi per l’assegnazione dei punteggi alle offerte nel caso di gara pubblica (vedi tabella 2.1);
- la determinazione dell’alternativa più adeguata nel caso in cui la fornitura non avvenga tramite il meccanismo della gara pubblica (ad esempio perché l’importo globale della fornitura è inferiore al limite oltre il quale la norma impone l’obbligo della gara).

Si noti che con la dizione “alternativa disponibile” si può intendere sia un singolo prodotto (nel caso si voglia scegliere tra due o più prodotti software ben definiti) che una soluzione informatica complessiva (nel caso si stia intraprendendo un progetto di realizzazione di un intero sistema informatico). Le indicazioni riportate nel seguito valgono, con alcune eccezioni, in entrambi i contesti.

Nel seguito ci si focalizza in particolare sulla comparazione tra l’acquisizione di programmi informatici di tipo proprietario (alternativa “c” della Direttiva) e l’acquisizione di programmi informatici a codice sorgente aperto (alternativa “d”), in quanto:

- ciò rappresenta l’opzione più innovativa e di più complessa valutazione tecnico-economica, data la limitata conoscenza ed esperienza a oggi disponibile nell’ambito della pubblica amministrazione sul software OS;
- le alternative “c” e “d” della Direttiva sono, sul piano organizzativo-strategico, tutto sommato simili (in entrambi i casi si sceglie “buy” anziché “make”) e la convergenza sulla scelta finale può essere, come vedremo, strutturata.
- Il confronto tra le alternative “c” e “d” rientra più direttamente nell’ambito dei compiti assegnati a questo gruppo di lavoro. L’opzione “b” è approfondita dal gruppo di lavoro del CNIPA che segue specificatamente la tematica del riuso.

La necessità di definire una metodologia di valutazione specifica nasce dal fatto che, per le loro rispettive nature, non sempre è corretto valutare con lo stesso metro un prodotto software di tipo proprietario e un prodotto software open source: può accadere, con riferimento ad alcune particolari caratteristiche dei prodotti software, che usando un metodo di valutazione inadatto si finisca per mettere in comparazione oggetti o qualità non confrontabili.

Per chiarire quanto appena detto, in tabella 2.2 vengono riepilogate alcune delle differenze più importanti tra i prodotti software di tipo proprietario e quelli di tipo open source.

**Tabella 2.2: principali differenze software proprietario – software open source**

	<b>Prodotti Proprietari</b>	<b>Prodotti Open Source</b>
<b>Fornitore del software e dei servizi di assistenza</b>	Un’impresa	Un distributore o una comunità
<b>Evoluzione del prodotto</b>	Determinato dalla strategia del fornitore	Determinato dalle funzionalità richieste al prodotto
<b>Sviluppatori</b>	Numero limitato, pagati dal fornitore	Possono essere anche molto numerosi, e variano da impiegati stipendiati a volontari
<b>Stabilità di versione</b>	Determinata da motivi commerciali	Determinata dalle richieste degli utenti
<b>Utenti</b>	Normalmente non organizzati. Ogni utente mantiene indipendentemente contatto con il fornitore	Organizzati in comunità virtuali. In stretto contatto con gli sviluppatori.

Le differenze di cui sopra (e le altre differenze approfondite in altre parti del presente documento) implicano che i tradizionali indicatori utilizzati per misurare ad esempio la funzionalità, l'affidabilità, l'usabilità dei prodotti software potrebbero non essere adeguati (al limite, non applicabili) anche ai prodotti software di tipo open source.

La metodologia proposta è rappresentata in figura 1.2, con evidenziate:

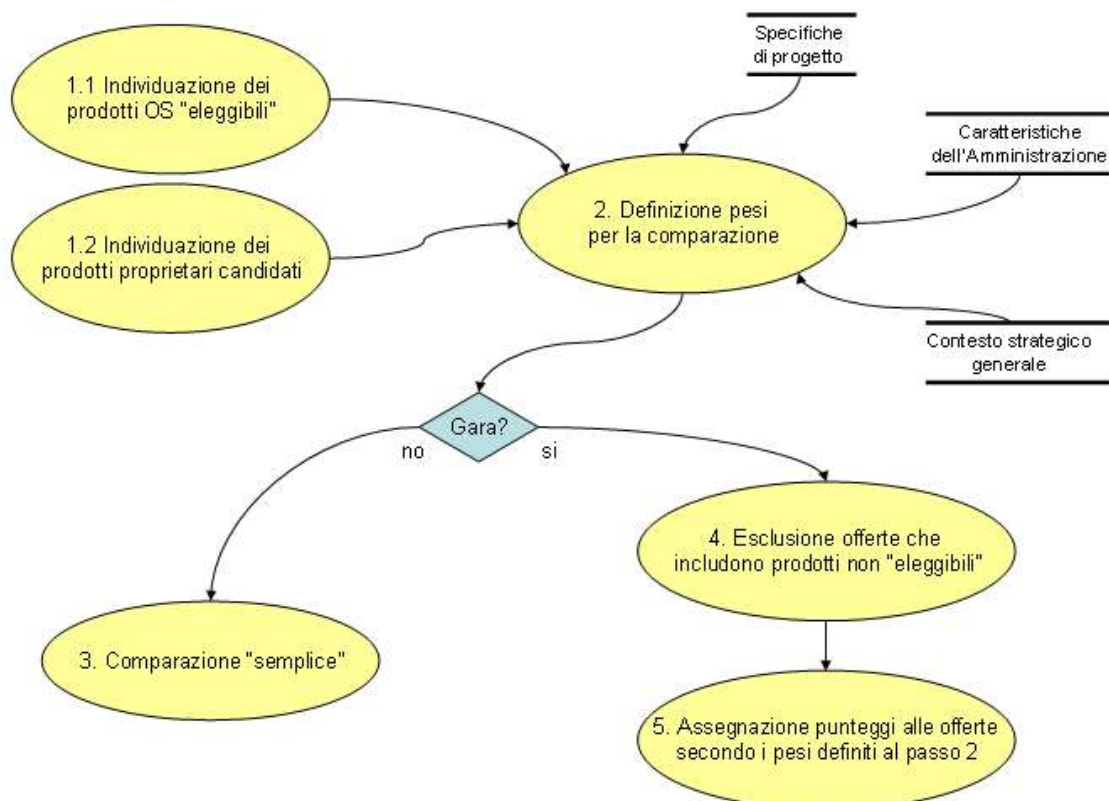
- le distinte attività che la compongono;
- i flussi di dati tra le varie attività.

Ogni attività è poi descritta in dettaglio nei paragrafi che seguono.

Si ribadisce ancora una volta che la metodologia proposta si configura semplicemente come una serie di indicazioni suggerite per rendere più efficiente e sicuro il processo di comparazione delle alternative disponibili.

Si noti inoltre che l'impegno profuso da un'amministrazione nella valutazione di prodotti software (o di una soluzione complessiva) deve essere proporzionale all'importanza che tali prodotti (o tali soluzioni) rivestono per l'amministrazione stessa.

In altre parole, l'applicazione completa di una metodologia complessa quale quella proposta ha senso solo in progetti di una certa rilevanza, e all'interno di questi solo per le parti di maggiore importanza. Ove la scelta tra più alternative abbia scarsa rilevanza nell'ambito del progetto complessivo, l'amministrazione potrà seguire solo le indicazioni principali (al limite nessuna) della metodologia proposta.



**Figura 2.2: schema della metodologia**

### 2.3.1. Individuazione dei prodotti OS “eleggibili”

Il numero e la dinamica (intesa come nascita, evoluzione, scomparsa) dei prodotti software di tipo open source attualmente esistenti sul mercato dello IT è tale che una qualunque ipotesi d'utilizzo dovrebbe essere ragionevolmente preceduta da un primo “filtro”, che elimini tutti quei prodotti che per bassa maturità, scarso supporto, insufficiente qualità e altre considerazioni è ragionevole non prendere in considerazione.

Ciò equivale ad esempio alla pratica di escludere, nel mondo del software proprietario, l'uso di prodotti ancora in versione “beta”, perché presumibilmente non ancora stabili.

Le principali caratteristiche sulla base delle quali escludere o meno un prodotto open source sono le seguenti:

- Età. In genere il primo anno di vita di un prodotto open source è il più critico, e la maggior parte dei nuovi progetti open source fallisce nel corso del primo anno, per cui scegliere prodotti con due anni o più di vita normalmente riduce il rischio di scegliere un software destinato a scomparire.
- Motivazione degli sviluppi. Se un progetto open source nasce e si evolve perché c'è una reale opportunità di mercato da sfruttare, esso ha maggiori probabilità di successo; viceversa, se il progetto nasce da motivazioni individuali, le sue possibilità sono minori, e il rischio di fallimento più alto.
- Organizzazione della comunità degli sviluppatori, in termini di numero di sviluppatori attivi, gerarchia esistente, struttura di controllo e gestione delle versioni, disponibilità di servizi a valore aggiunto quali consulenza, formazione e integrazione.
- Penetrazione sul mercato. La diffusione del prodotto sul mercato, nonché il suo posizionamento nei confronti dei prodotti concorrenti, sono indice di successo e riducono il fattore di rischio. È peraltro molto difficile ottenere un dato oggettivo di diffusione di un prodotto open source, perché per loro natura questi software possono essere scaricati e installati senza che ne resti traccia (non esistono fatture o altri documenti del genere). Un indicatore significativo è, ad esempio, il numero di distribuzioni commerciali (es. Red Hat) in cui il prodotto sotto esame è incluso.
- Disponibilità di documentazione, sia in termini di tipologia (manuali, FAQ, HowTo, eccetera) che in termini di numero e qualità;
- Integrabilità del prodotto con altri software (disponibilità di API, adozione di standard riconosciuti, modularità, ecc.)

Per dare un voto a ogni caratteristica del prodotto open source in esame, si può procedere raccogliendo informazioni dai siti internet dedicati all'open source, quali ad esempio SourceForge<sup>64</sup>, Freshmeat<sup>65</sup>, FSF<sup>66</sup>, Icewalker<sup>67</sup>, che mettono a disposizione sia schede tecniche che valutazioni degli utenti e dati d'utilizzo.

---

<sup>64</sup> <http://sourceforge.net>

<sup>65</sup> <http://freshmeat.net>

<sup>66</sup> <http://www.fsf.org>

<sup>67</sup> <http://www.icewalk.com>

Generalmente, inoltre, ogni prodotto open source dispone di un sito web ufficiale (OS official project web site), ove si possono reperire sia le informazioni sugli sviluppatori e l'evoluzione del prodotto, sia la relativa documentazione (FAQ, Howto, eccetera). È buona norma comunque, prima di assegnare un voto alle caratteristiche sopra elencate, raccogliere informazioni da più fonti, eventualmente verificando i dati tra loro discordanti. Si deve inoltre identificare attentamente la versione del prodotto in esame, perché nel campo dei prodotti open source, a causa della rapida evoluzione, ciò che è vero per una versione può non esserlo per la versione successiva.

Per la sua complessità, si consiglia, ove necessario, che il primo passo della metodologia venga compiuto anche basandosi sulle indicazioni del Centro di Competenza presso il CNIPA: periodicamente quest'ultimo effettuerà rilevazioni e valutazioni di prodotti open source, e poi metterà i risultati delle sue attività a disposizione delle amministrazioni interessate. A titolo di riferimento, il progetto comunitario IDA offre già un servizio di questo genere<sup>68</sup>: sul sito web della Commissione europea esiste una lista di programmi OS suggeriti, con l'evidenza di quelli valutati "maturi".

Resta inteso che, ove un'amministrazione abbia già evidenza e informazioni sufficienti su un particolare prodotto open source (ad esempio perché ha partecipato sin dall'inizio al progetto, o addirittura lo ha promosso), non è obbligata a servirsi delle indicazioni del Centro di Competenza del CNIPA: le valutazioni di quest'ultimo sono offerte a titolo di supporto facoltativo per le amministrazioni che non dispongono di mezzi e risorse sufficienti e necessitano perciò di ausilio nell'individuazione dei prodotti "eleggibili".

### **2.3.2. Individuazione dei prodotti proprietari candidati**

Apparentemente questa attività è simile alla precedente. Si individuano anche in questo caso alcune caratteristiche sulla base delle quali escludere o meno un prodotto proprietario dalla lista di possibili "candidati all'uso". Le più importanti tra queste caratteristiche sono:

- Maturità del prodotto. È opportuno prendere in considerazione solo prodotti che abbiano raggiunto un grado di stabilità superiore a una soglia minima, evitando ad esempio prototipi, versioni beta, software sperimentali. Il numero di versione di un prodotto proprietario è un primo indicatore della sua maturità (ma non è l'unico da prendere in considerazione).
- Condizioni finanziarie e "vitalità" del produttore. Anche nel recente passato si sono verificati casi di fornitori che sono usciti dal mercato IT per fallimento, che sono stati acquisiti da fornitori più grandi, o che hanno dovuto per qualche motivo rinunciare ad alcuni loro settori d'attività. Sembra probabile che casi del genere continueranno a verificarsi in un mercato variabile, altamente competitivo e non facilmente prevedibile come l'IT. In caso di uscita dal mercato di un fornitore, i prodotti di cui esso è proprietario rischiano di non essere più supportati, o che la loro evoluzione venga abbandonata. Per questo motivo, è opportuno escludere dalla lista di prodotti candidati i software proprietà di fornitori le cui condizioni societarie e/o finanziarie facciano sospettare un'uscita dal mercato.
- Penetrazione sul mercato. Anche per i prodotti proprietari, l'indice di penetrazione sul mercato, nonché il posizionamento nei confronti dei prodotti concorrenti, sono indice di successo e riducono il fattore di rischio. Questa informazione, per i

<sup>68</sup> <http://europa.eu.int/ISPO/ida/jsps/index.jsp?fuseAction=showChapter&chapterID=452&preChapterID=0>

prodotti proprietari, è facilmente rintracciabile (anche se non è opportuno basarsi solo sui dati pubblicizzati dal fornitore).

- Caratteristiche della rete di distribuzione e di supporto. Molti prodotti proprietari appartengono ad aziende straniere (soprattutto statunitensi) che si appoggiano a distributori e partner locali per le attività di vendita, installazione e supporto. In questi casi è opportuno prendere in considerazione (o escludere) il prodotto non solo sulla base della valutazione del produttore, ma anche a seguito di valutazione della rete dei distributori e dei partner locali: esistono infatti diversi buoni prodotti proprietari distribuiti e supportati in modo inadeguato sul mercato italiano.

La valutazione delle caratteristiche di cui sopra viene svolta in maniera diversa da quanto previsto nell'attività 1.1, ancora una volta a causa della diversa natura delle due tipologie di prodotti software a confronto (proprietary e open source).

Mentre infatti l'identificazione e la valutazione dei software open source può essere svolta esaminando direttamente i prodotti stessi (e il corrispondente progetto di sviluppo), normalmente non tutte le informazioni necessarie per una valutazione sono disponibili per i software proprietari (il codice sorgente, ma anche informazioni sugli sviluppatori, le strategie di evoluzione futura, i giudizi degli utenti, le anomalie già segnalate, ecc.).

Per questi motivi, la valutazione dei prodotti proprietari si basa usualmente sulle indicazioni rilasciate dagli stessi fornitori. Sono peraltro da prendere in considerazione anche gli articoli su riviste tecniche e le opinioni espresse dagli analisti di mercato, ed è comunque possibile effettuare ricerche di prodotti concorrenti tramite generici motori su Internet o strumenti di ricerca su siti specializzati (es. KnowledgeStorm<sup>69</sup>).

Si noti che normalmente sono i fornitori stessi a compiere l'attività di segnalazione dei propri prodotti, tramite pubblicità, eventi, dimostrazioni pubbliche, presentazioni mirate presso il singolo cliente, eccetera. Questa è un'altra importante differenza esistente tra i software proprietari e i software open source, che generalmente non vengono pubblicizzati (fa eccezione Linux, ormai stabilmente parte dell'offerta commerciale "strategica" di alcune grandi aziende, in primo luogo IBM).

Le grandi amministrazioni pubbliche, considerate potenziali buoni clienti dalla maggior parte dei fornitori IT, non hanno dunque necessità di compiere in proprio rilevazioni di prodotti software proprietari: sono i fornitori stessi a portar loro tutte le informazioni necessarie, anche sotto forma di comparazioni tecniche tra i loro prodotti e quelli dei concorrenti.

Al contrario, per le amministrazioni di dimensioni minori, specie le locali, acquisire informazioni complete e aggiornate sui prodotti proprietari disponibili può rappresentare un problema, essendo tali amministrazioni usualmente fuori dal giro d'interesse dei fornitori (e dunque trascurate dai loro commerciali).

Per superare questo problema possono essere d'aiuto anche le iniziative condotte da strutture centrali, ad esempio il CNIPA, quali gli accordi strategici con i più importanti fornitori IT, i contratti quadro per la fornitura di licenze d'uso, le Convenzioni, i marketplace elettronici, ecc.

---

<sup>69</sup> <http://www.knowledgestorm.com>



### 2.3.3. Definizione pesi per la comparazione

Come evidenziato in figura 2.1, questa attività ha come input:

- le specifiche di progetto (sia le specifiche tecnologiche del sistema informatico da realizzare che ulteriori specifiche progettuali quali scadenze, impegno complessivo, modalità di messa in produzione, tempo di vita previsto per il sistema da realizzare, eccetera);
- le caratteristiche dell'amministrazione, in termini di organizzazione interna, schema dei processi, dimensione, ma anche con riguardo alle caratteristiche tecnologiche degli attuali sistemi informatici con cui il sistema da realizzare dovrà interagire;
- il contesto generale, con particolare riferimento alle attese dell'utenza interna ed esterna (cittadini), alle motivazioni "strategiche" che hanno dato luogo al progetto, alla normativa di riferimento (con attenzione alla prevista evoluzione della stessa), alla collaborazione con altre amministrazioni e ad eventuali vantaggi per queste ultime (es. riuso).

Come output, l'attività fornisce i "pesi" dei vari criteri di comparazione (vedi tabella 2.2) che verranno utilizzati dalle attività poste a valle nel flusso previsto dalla metodologia.

L'attività può essere svolta con l'ausilio di un questionario o di tabelle di riferimento, su cui assegnare l'importanza che i singoli criteri di comparazione rivestono nell'ambito dell'amministrazione e del progetto in corso. Le tabelle di riferimento forniranno il "peso" numerico corrispondente all'importanza assegnata dal compilatore.

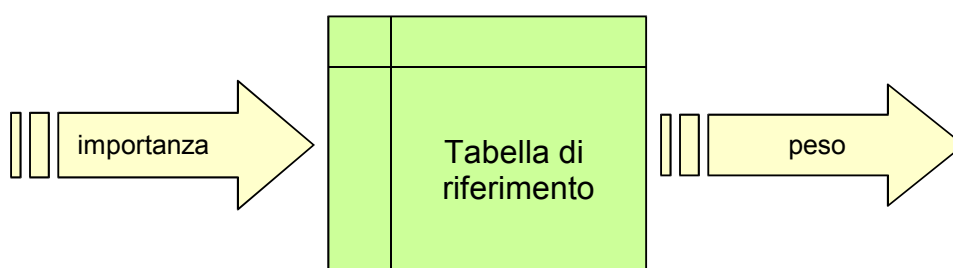


Figura 2.3: calcolo del "peso"

In tabella 2.3, lo schema di un possibile questionario.

Con tale strumento il responsabile di progetto (o i responsabili delle varie aree coinvolte) assegna un grado di importanza ai singoli criteri sulla base degli elementi elencati tra gli input, ma anche affidandosi alla propria percezione ed esperienza professionale. È opportuno inoltre che il compilatore del questionario specifichi nel campo Note la motivazione del grado di importanza assegnato, oppure riporti il riferimento alla specifica di progetto (o ad altro input, se identificato) che ha portato all'assegnazione (ciò consentirà di mantenere traccia delle valutazioni effettuate e delle decisioni prese).

Tabella 2.3: schema d'esempio di questionario

Criterio	Importanza (ininfluente, bassa, media, alta, molto alta)	Note
Funzionalità		
Affidabilità del prodotto/soluzione		
Affidabilità del fornitore		
Usabilità		
Prestazioni		

Scalabilità		
Flessibilità		
Riusabilità		
Sicurezza		
Disponibilità dei sorgenti per ispezioni e tracciabilità		
Indipendenza da tecnologie proprietarie o di unico fornitore		
Uso di formati e standard aperti		
Tempistica progetto (scadenze)		
Garanzie legali		
Livello di supporto/manutenzione		
Compatibilità con l'ambiente tecnologico dell'amministrazione		
Qualità della formazione		
TCO		
Costo di uscita		

Si noti che non tutti i criteri sono indipendenti tra loro (ad esempio la Sicurezza e la Disponibilità dei sorgenti sono legate) e non tutti i criteri sono significativi in tutti i possibili contesti. Per questo motivo il questionario va compilato solo per le voci che hanno senso. Analogamente, il responsabile di progetto (o i responsabili delle varie aree coinvolte) può aggiungere altri criteri oltre a quelli elencati in tabella (si ribadisce che si tratta solo di un esempio).

Si noti inoltre che i criteri economici TCO e Costo di uscita (le cui definizioni sono date al paragrafo ) dovrebbero avere, nella generalità dei casi, importanza alta o molto alta. È difatti da considerarsi un'eccezione il caso di un progetto in cui gli aspetti economici abbiano scarsa rilevanza.

Si riporta nel seguito un esempio di compilazione del questionario, in cui la relazione tra i pesi e l'importanza assegnata è lineare, vale a dire in cui esiste la corrispondenza “ininfluente” = 0, “bassa importanza” = 1, “media importanza” = 2, “alta importanza” = 3, “importanza molto alta” = 4.

**Tabella 2.4: esempio compilazione questionario**

Criterio	Importanza	Peso	Note
Funzionalità	Molto alta	4	La strategia dell'amministrazione e l'impostazione generale del progetto puntano a premiare soprattutto gli aspetti funzionali della soluzione
Affidabilità del prodotto	Media	2	Requisito di progetto (riferimento al SRD <sup>70</sup> )
Affidabilità del fornitore	Media	2	Dalla strategia generale dell'amministrazione
Usabilità	Alta	3	Requisito di progetto (riferimento al SRD)
Prestazioni	Ininfluente	0	Non esistono specifiche di progetto a riguardo
Scalabilità	Ininfluente	0	Non esistono specifiche di progetto a riguardo
Flessibilità	Ininfluente	0	Non esistono specifiche di progetto a riguardo
Riusabilità	Ininfluente	0	Il contesto applicativo del progetto, molto specifico, è tale per cui non esistono altre amministrazioni interessate al riuso
Sicurezza	Alta	3	Richiesta dalla “missione” e dalla strategia complessiva dell'amministrazione
Disponibilità dei sorgenti per ispezioni e tracciabilità	Alta	3	Richiesta dalla “missione” e dalla strategia complessiva dell'amministrazione
Indipendenza da tecnologie proprietarie o di unico fornitore	Media	2	Non espressamente prevista dalle specifiche di progetto, ma parte della strategia complessiva dell'amministrazione
Uso di formati e standard aperti	Alta	3	Requisito di progetto (riferimento al SRD)
Tempistica progetto (scadenze)	Bassa	1	Le specifiche di progetto prevedono ampi margini sulle scadenze

<sup>70</sup> System Requirement Document

Garanzie legali	Bassa	1	L'amministrazione non richiede particolari garanzie legali sul sistema da realizzare
Livello di supporto/manutenzione	Media	2	Nelle specifiche di progetto non sono definiti particolari livelli di servizio
Compatibilità con l'ambiente tecnologico dell'amministrazione	Ininfluyente	0	Non sono previste interazioni tra il sistema da realizzare e la restante infrastruttura IT dell'amministrazione
Qualità della formazione	Ininfluyente	0	Nel progetto non è prevista attività di formazione
TCO	Molto alta	4	Il costo del progetto è un criterio fondamentale di scelta
Costo di uscita	Alta	3	La strategia dell'amministrazione non prevede espressamente un futuro abbandono della tecnologia, tuttavia questo criterio viene considerato di alta importanza

### 2.3.4. Esclusione offerte con prodotti “non eleggibili”

Questa attività, che è opportuno svolgere in maniera separata nel caso di gara pubblica (non necessariamente nel caso di trattativa privata), consiste nell'eliminare dalle offerte ricevute tutte quelle che includono prodotti software valutati, nelle attività 1.1 e 1.2, non sufficientemente maturi e consolidati.

Allo stesso modo, vanno escluse tutte quelle offerte che includono prodotti open source e proprietari i cui modelli di licenza risultino incompatibili tra loro. Ciò per evitare il rischio di problemi legali nell'uso (e nell'evoluzione futura) della soluzione complessiva.

Si ribadisce quanto detto al paragrafo sull'uso facoltativo, da parte delle amministrazioni, delle indicazioni del Centro di Competenza CNIPA. Un'eventuale lista di prodotti software open source “eleggibili” fornita dal suddetto Centro dovrà essere intesa come strumento d'ausilio per le amministrazioni che, per insufficiente visibilità del mercato, hanno bisogno di una valutazione “terza” qualificata, ma non potrà prevalere sulle valutazioni compiute in proprio dalle amministrazioni, che mantengono comunque la propria autonomia decisionale e la relativa responsabilità di scelta.

### 2.3.5. Assegnazione punteggi

L'attività di assegnazione dei punteggi, da svolgersi nel caso di gara pubblica, consiste nel prendere in esame le varie offerte e assegnare un voto (ad esempio, da 1 a 10) a tutte le caratteristiche (criteri) delle offerte che non sono state giudicate influenti al punto 2.

Questa attività deve essere svolta con attenzione qualora siano a confronto prodotti software proprietari e prodotti software open source: come già detto, occorre essere certi di non mettere in comparazione oggetti o qualità non confrontabili.

Con riferimento alle tabelle 2.3 e 2.4, si evidenziano alcune differenze sostanziali, nella assegnazione dei punteggi, tra i prodotti proprietari e i prodotti open source.

- Funzionalità. Assegnare un punteggio alla funzionalità di prodotti proprietari significa normalmente eseguire un confronto tra le funzionalità richieste nelle specifiche di progetto e le funzionalità dichiarate dai fornitori dei prodotti. Ciò può essere fatto ad esempio utilizzando una check list come quella presentata di seguito.

	Prodotto1	Prodotto2	Prodotto3
Funzionalità 1	SI	NO	SI
Funzionalità 2	SI	SI	NO
...	SI	SI	NO

Funzionalità N	SI	SI	SI
Punteggio finale	10	9	8

Per prodotti open source, la verifica della presenza delle funzionalità richieste può non essere altrettanto immediata. Una funzionalità potrebbe infatti non essere presente in maniera nativa, ma essere comunque facilmente implementabile. Dunque l'indicatore tradizionale "funzionalità presente / funzionalità assente" è poco efficace, se non in aggiunta ad altri indicatori quali la velocità di implementazione, la capacità degli sviluppatori di aggiungere la funzionalità richiesta, la disponibilità di adeguati strumenti di sviluppo e debugging.

Operativamente, l'amministrazione può assegnare il punteggio in maniera inversamente proporzionale all'impegno stimato per implementare le funzionalità mancanti: tale stima può essere desunta dall'offerta stessa (se è un'offerta di servizi di integrazione), essere richiesta a un esperto "terzo", o espressamente alla comunità degli sviluppatori (questi ultimi rappresentano una fonte autorevole, in quanto il loro business usuale consiste proprio nell'estensione delle funzionalità di base del prodotto).

- Livello di supporto/manutenzione. Per i prodotti proprietari, l'assegnazione del punteggio consiste essenzialmente in un giudizio sulle modalità di supporto e le opportunità di evoluzione futura dichiarate dal fornitore. Al contrario, per i prodotti open source esistono più opzioni possibili (affidarsi a un fornitore di servizi, formare/organizzare un gruppo di supporto interno, iscriversi a una comunità di utenti). Dunque, nel caso dei prodotti open source, l'assegnazione del punteggio si baserà ad esempio sul grado di coinvolgimento del fornitore di servizi nel processo di sviluppo del prodotto (se è un main contributors, se e quanto ha investito nel progetto, ecc.). Esistono inoltre degli "indicatori di vitalità" dei prodotti open source disponibili in rete (ad esempio, il sito Freshmeat riporta indicatori di questo tipo): si noti che in alcuni casi (specie in prodotti molto specializzati) la bassa frequenza nei rilasci di nuove versioni non è un cattivo segnale, anzi è un indicatore positivo di stabilità e buon funzionamento del prodotto.
- Affidabilità. Anche per l'affidabilità, l'assegnazione del punteggio ai prodotti proprietari deve basarsi soprattutto su dati rilasciati dallo stesso fornitore, mentre per i prodotti open source sono disponibili fonti di altro tipo. Si cita ad esempio, come indicatore efficace, il trend delle segnalazioni di anomalie inviate dagli utenti di un prodotto sul sito web ufficiale del progetto.
- Prestazioni. Per i prodotti proprietari l'assegnazione del punteggio alla caratteristica "prestazioni" deve basarsi sui dati dichiarati dai fornitori (eventualmente integrati dalle indicazioni, se esistenti, di riviste tecniche o di analisti indipendenti). Per i prodotti open source sono a volte disponibili dati di prestazione sui siti ufficiali del progetto, oppure risultati di test di prestazione in configurazioni di carico "tipiche" eseguite da comunità di utenti. Esistono inoltre strumenti per misurare le prestazioni di diversi software open source in contesti simulati ma realistici.
- Garanzie legali. Per i software proprietari l'assegnazione del punteggio deriva essenzialmente dal giudizio sull'End User License Agreement (EULA) offerto dal fornitore. In generale dovrebbero meritare bassi punteggi quei prodotti i cui EULA includono clausole legalmente pericolose per il cliente (ad esempio, alcuni EULA riservano al fornitore il diritto di accesso ai contenuti dei PC del cliente o a

informazioni private dello stesso, o limitano l'uso che il cliente è autorizzato a fare del prodotto, o prevedono controlli e ispezioni) e meritare alti punteggi i prodotti i cui EULA includano garanzie aggiuntive (ad esempio un certo grado di copertura contro eventuali danni provocati dall'uso del prodotto). A questo proposito, si noti che molto difficilmente un fornitore garantisce completi rimborsi a danni provocati dall'uso di suoi prodotti: normalmente viene offerto solo un rimborso del prezzo della licenza.

Per i software open source, l'assegnazione del punteggio avverrà tenendo conto della presenza di eventuali azioni legali che possano influenzare l'evoluzione del prodotto, oppure del grado di allineamento del tipo di licenza con la normativa in vigore o in iter di approvazione (ad esempio, la legge sulla proprietà intellettuale, la legge sulla brevettabilità del software, ecc.), e soprattutto della garanzia che la comunità o l'integratore offrono al cliente nei riguardi di azioni legali intraprese da terzi nei loro confronti (liability).

- Sicurezza. Per i prodotti proprietari, l'assegnazione del punteggio all'aspetto sicurezza può avvenire sulla base delle informazioni messe a disposizione dal fornitore (eventualmente sulla scorta di certificazioni a cura di enti terzi). Per i software open source, la disponibilità del codice può permettere un esame "interno" del prodotto (non necessariamente a cura del cliente). Esistono degli strumenti automatici (es. RATS, Flawfinder) che scandiscono il sorgente di un software alla ricerca delle più comuni cause di debolezze di sicurezza, quali ad esempio i meccanismi di assunzione di privilegio, i controlli di input, ecc.
- TCO. Per il costo totale di possesso, il punteggio viene assegnato sulla base di una formula matematica del tipo:

$$\text{Punteggio} = C_1 * (P_{\min} + P_{\max} - P) / P_{\max} \quad F2.1$$

Dove:

- $P_{\min}$  è il TCO minimo tra le alternative,
- $P_{\max}$  è il TCO massimo tra le alternative,
- $C_1$  è il massimo punteggio assegnabile.

Il costo totale di possesso delle varie alternative, nel suo valore numerico (in euro), deve essere ricavato, per quanto possibile, direttamente dalle offerte a confronto. Per le voci che compongono il TCO e che non sono esplicitamente riportate nell'offerta del fornitore, l'amministrazione dovrà ricorrere a stime. Ricordiamo che le voci principali del TCO di una soluzione informatica sono:

- i costi di acquisizione dell'hardware e dei mezzi trasmissivi (acquisto, noleggio, aggiornamento);
- i costi di acquisizione del software (licenze d'uso, costi di sviluppo o di personalizzazione);
- i costi di gestione (installazione, deployment, backup, supporto, manutenzione dell'hardware e del software, usura delle parti consumabili);

- i costi di amministrazione (rapporti con i fornitori, approvvigionamento);
- i costi relativi all'operatività degli utenti (formazione, affiancamento, assistenza, configurazione delle applicazioni, gestione dei dati, consulenze);
- i costi dell'inattività della soluzione (per guasti, anomalie, fermo del sistema, ecc.).

Si noti che i costi relativi all'hardware possono essere significativi anche nei confronti tra alternative riguardanti solo prodotti software. Ad esempio, la scelta di un certo prodotto software potrebbe comportare la necessità di un aggiornamento dell'hardware per garantire le prestazioni necessarie. O ancora, un certo prodotto software potrebbe non essere compatibile con l'hardware attualmente a disposizione, per cui la scelta del suddetto software potrebbe implicare l'acquisto di adattatori, espansioni o addirittura la sostituzione dell'hardware.

Si noti inoltre che per calcolare il TCO di una soluzione informatica occorre prima fissare il tempo di vita della soluzione stessa. Ad esempio, stabilendo che il sistema informatico da realizzare debba restare operativo tre anni, i costi di gestione, amministrazione e in generale tutti i costi periodici andranno calcolati con riferimento al suddetto intervallo di tempo.

L'applicazione della formula F2.1 non viene influenzata dalla presenza di prodotti software open source: assumono semplicemente diversa rilevanza alcune componenti di costo quali le licenze d'uso (che potrebbero anche essere assenti).

- Costo di uscita. Il punteggio al costo di uscita viene assegnato, come nella formula F1.1, sulla base di una proporzionalità inversa tra costo e punteggio: anche in questo caso, infatti, l'alternativa con costo di uscita minore avrà il punteggio massimo e viceversa.

Il costo di uscita delle varie alternative, nel suo valore numerico (in euro) deve essere ricavato, se possibile, direttamente dalle offerte a confronto, per cui i fornitori dovranno essere invitati, nel capitolato di gara, a indicare anche i costi di uscita delle soluzioni che offrono. Per le voci che compongono il costo di uscita e che non sono esplicitamente riportate nell'offerta del fornitore, l'amministrazione dovrà ricorrere a stime.

Ricordiamo che le voci principali del costo di uscita di una soluzione informatica sono:

- i costi della conversione di dati memorizzati in formato proprietario;
- costi di dismissione e smaltimento di apparati hardware e/o di trasmissione da abbandonare;
- i costi di intervento su eventuali sistemi “terzi” che interfacciano la soluzione informatica da abbandonare;
- i costi di verifica e test di compatibilità anche sui sistemi “terzi” di cui sopra su cui (in teoria) non è necessario intervenire;

- costi di assistenza e “riaddestramento” degli utenti della soluzione informatica da abbandonare;
- i costi di inattività legati all’abbandono della soluzione informatica (disinstallazione, riconfigurazione postazioni di lavoro, risoluzione problemi, downtime, ecc.).

Una volta assegnati i punteggi a tutte le N caratteristiche, essi dovranno essere combinati secondo la formula:

$$\text{Punteggio complessivo} = \sum_{i=1}^N (\text{PESO}_i * \text{PUNTEGGIO}_i) \quad \text{F2.2}$$

dove:

- $\text{PESO}_i$  = peso assegnato al passo 2 alla caratteristica i-esima.
- $\text{PUNTEGGIO}_i$  = punteggio assegnato alla caratteristica i-esima.

Per illustrare meglio il processo, di seguito viene presentato un esempio di comparazione tra 3 alternative. Come pesi vengono utilizzati quelli definiti nell’esempio del paragrafo (eliminando quelli definiti “ininfluenti”).

Anzitutto viene calcolato il TCO delle tre alternative sulla base, come detto in precedenza, delle offerte presentate e di stime per le voci di costo non presenti. Ai tre TCO così calcolati viene poi applicata la formula F2.1 con una costante  $C_1$  pari a  $10^{71}$ , ottenendo in tal modo il punteggio corrispondente.

Viene poi calcolato il costo di uscita delle tre alternative, e si applica nuovamente la formula F2.1 per ottenere il punteggio corrispondente.

**Tabella 2.5: calcolo dei punteggi dei criteri economici**

	<b>Prodotto 1</b>	<b>Prodotto 2</b>	<b>Prodotto 3</b>
TCO (euro)	2.700.000	1.450.000	2.290.000
Punteggio TCO	5,4	10	6,9
Costo di uscita (euro)	305.000	120.000	175.000
Punteggio Costo di uscita	3,9	10	8,2

A questo punto i punteggi vengono sommarizzati secondo la formula F2.2, ottenendo i punteggi complessivi di tabella 2.6.

<sup>71</sup> L’ipotesi di lavoro  $C_1 = 10$  deve intendersi come un semplice esempio e non come una regola assoluta. Esistono in letteratura diversi studi relativi alla definizione ottimale del punteggio massimo, che continuano a essere validi anche a seguito della Direttiva OS (e a cui si consiglia perciò di fare riferimento). In generale, la scelta del punteggio massimo deve essere fatta in modo che il punteggio di ogni alternativa cresca se decresce il TCO/Costo di uscita, e crescono complessivamente gli altri criteri di valutazione. In particolare, devono essere rispettate le seguenti asserzioni:

- P.ggio complessivo(TCO min, qualità min) < P.ggio complessivo (TCO max, qualità media);
- P.ggio complessivo(TCO max, qualità max) >= P.ggio complessivo (TCO min, qualità media)

**Tabella 2.6: calcolo dei punteggi complessivi**

Criterio	Peso	Prodotto 1	Prodotto 2	Prodotto 3
Funzionalità	4	7	9	10
Affidabilità del prodotto	2	6	6	10
Affidabilità del fornitore	2	7	9	8
Usabilità	3	8	7	6
Sicurezza	3	5	8	7
Disponibilità dei sorgenti per ispezioni e tracciabilità	3	10	0	0
Indipendenza da tecnologie proprietarie o di unico fornitore	2	10	5	7
Uso di formati e standard aperti	3	10	8	7
Tempistica progetto (scadenze)	1	- <sup>72</sup>	-	-
Garanzie legali	1	6	9	8
Livello di supporto/manutenzione	2	7	10	9
TCO	4	5,4	10	6,9
Costo di uscita	3	3,9	10	8,2
<b>Punteggio complessivo</b>		<b>226,4</b>	<b>244</b>	<b>228,2</b>

A seguito della comparazione, risulta tecnicamente da preferire la seconda alternativa (Prodotto 2).

### 2.3.6.Comparazione “semplice”

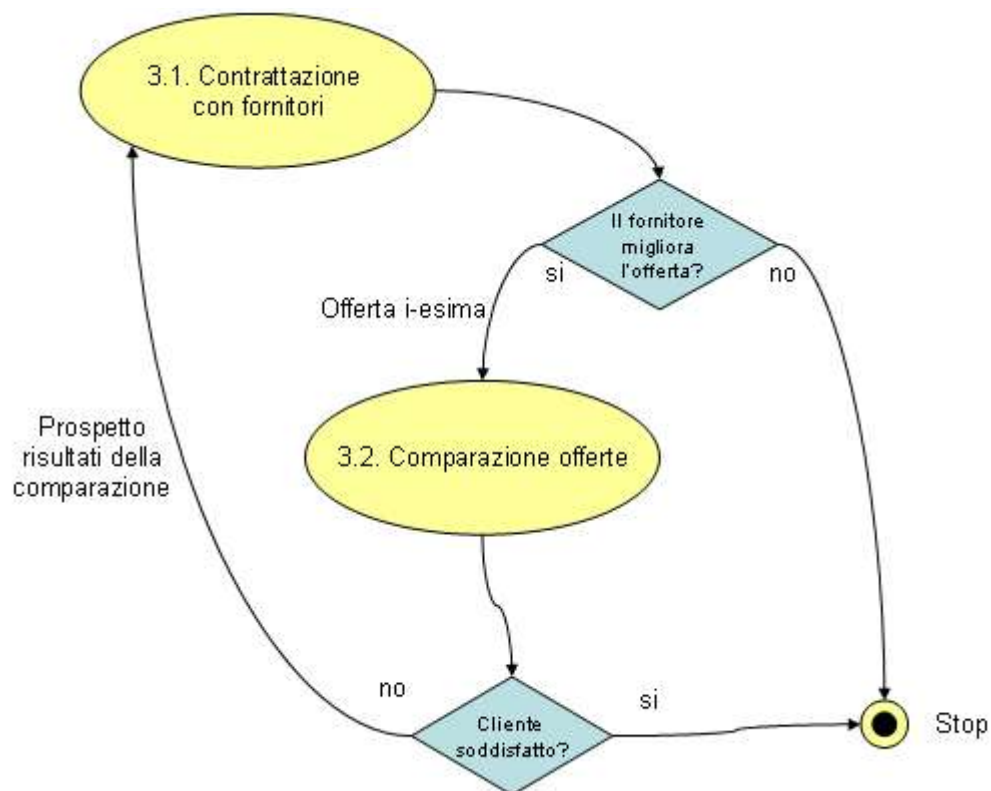
Questa attività, concettualmente analoga alla precedente, deve essere svolta nel caso l’acquisizione di prodotti o soluzioni informatiche da parte dell’amministrazione non avvenga tramite il meccanismo della gara pubblica, ma ad esempio tramite una trattativa privata con uno o più fornitori.

In questo caso la comparazione avviene in modalità meno strutturata rispetto al caso precedente, poiché laddove il meccanismo della gara “congelava” i termini delle offerte da mettere a confronto, in quanto stabiliti per iscritto nella risposta al bando di gara, in questo caso alcune caratteristiche dell’offerta (in primo luogo il prezzo, ma anche le garanzie e i livelli di supporto) possono essere modificate più volte a seguito di contrattazione tra l’amministrazione e i fornitori.

In caso di trattative private (singole o multiple, ovvero con più fornitori), il meccanismo di comparazione prevede diversi “ricicli”: a ogni singola iterazione, il risultato del confronto tra le offerte i-esime può essere utilizzato come base di negoziazione per ottenere condizioni migliori dal fornitore. Lo schema è rappresentato dal seguente diagramma, “di secondo livello” rispetto al diagramma principale della metodologia (si tratta dello schema interno dell’attività 3 del diagramma principale).

<sup>72</sup> Non viene assegnato un punteggio a questa voce perché si tratta di una comparazione di prodotti e non di





**Figura 2.4: comparazione semplice**

Il meccanismo di pesi, criteri e tabelle descritto nel paragrafo precedente può essere impiegato in maniera sostanzialmente analogo, a ogni iterazione, nella sottoattività 3.2. È tuttavia ragionevole, dato che la sottoattività 3.2 deve essere ripetuta più volte, limitarsi a considerare i criteri più importanti. Il riciclo termina quando i fornitori dichiarano di non poter ulteriormente migliorare l'offerta, oppure quando l'amministrazione si ritiene soddisfatta di quanto ottenuto.

La disponibilità o meno di prodotti open source non ha grande impatto su questo meccanismo, se non per il fatto che in genere tale disponibilità aumenta il numero di alternative a disposizione, e rende quindi più favorevole la posizione del cliente nella trattativa. Ad esempio, si sono verificati casi in cui un importante leader di mercato ha concesso sconti molto superiori ai suoi standard commerciali perché il cliente ha prospettato una migrazione dei suoi sistemi a prodotti open source.

Di fatto, il diagramma di figura 2.4 rispecchia l'attuale sistema di trattativa privata utilizzato dalle amministrazioni, e ne rappresenta quindi solo un modello formalizzato.

Si noti che il termine "fornitore" può riferirsi sia a un produttore di software proprietario, sia a un distributore di un software open source, sia a un integratore di sistemi in grado di offrire entrambe le tipologie di software.

### **2.3.7. Il ruolo del CNIPA**

Sembra utile riepilogare, in conclusione, i passi della metodologia appena descritta in cui è coinvolto a vario titolo il CNIPA.

- Redazione dello Studio di Fattibilità e del Capitolato Tecnico. Come detto, questa attività è a carico delle amministrazioni (che eventualmente la affidano a società terze). Il CNIPA può eventualmente offrire un ausilio alle amministrazioni, ad

esempio mettendo a disposizione di queste ultime Studi di Fattibilità e Capitolati ove vengano applicate le indicazioni della Direttiva, da usare come modelli.

- Formulazione del parere. Come previsto dalla Direttiva, in sede di rilascio del parere di cui all'art. 8 del decreto legislativo 12 febbraio 1993, n. 39, il CNIPA valuterà le comparazioni di soluzioni alternative e le relative analisi costi/benefici eseguite dall'amministrazione e riportate nello Studio di Fattibilità.
- Individuazione dei prodotti software (OS o proprietari) candidati. Con riferimento a quanto espresso nei paragrafi e , il Centro di Competenza CNIPA fornirà, alle amministrazioni che le richiedano, ogni informazione in suo possesso utile a eseguire l'attività in questione, sia per quanto riguarda i prodotti software open source (eventualmente fornendo una lista aggiornata di "prodotti candidati") che i prodotti proprietari.
- Supporto all'intero processo. Almeno in un primo periodo transitorio, il Centro di Competenza del CNIPA potrà fornire, dietro richiesta, indicazioni, chiarimenti e supporto alle amministrazioni in tutte le fasi del processo. Le modalità con cui il supporto verrà fornito sono descritte nel capitolo relativo al Centro di Competenza

Alle attività di cui sopra si aggiunge un impegno iniziale di sperimentazione e raffinamento della metodologia stessa. Essa infatti costituisce ancora solo uno studio di tipo teorico. La metodologia verrà applicata su casi di prova allo scopo di verificarne l'efficacia ed eventualmente di metterla a punto per renderla usabile e adatta ai reali progetti della pubblica amministrazione.

## **2.4.Processo di valutazione nel caso di sviluppo ad hoc.**

La componente di sviluppo di software custom ha nella PA un peso economico particolarmente rilevante. Le realizzazioni ad hoc sono tipicamente molto costose a causa di una serie di fattori che le caratterizzano, tra i quali:

- difficoltà nel condividere i requisiti funzionali del sistema
- variabilità degli scenari normativi di riferimento;
- forte coinvolgimento richiesto alle strutture interne alla committenza;
- dipendenza dalla tecnologia;
- dipendenza dal fornitore;
- impiego di gruppi di progetto medio/grandi.

L'adozione di modelli di sviluppo OS può portare a mitigare le criticità elencate. Come si è detto nel capitolo introduttivo, la realtà del OS non è riferibile a un unico modello di sviluppo, ma piuttosto a una serie di best practice in costante evoluzione. E' proprio verso la verifica delle pratiche di sviluppo che la valutazione delle offerte dei fornitori dovrebbe concentrarsi. In particolare, si dovrebbero verificare i seguenti punti:

- utilizzo di standard aperti: è una pratica essenziale a prevenire la progettazione di soluzioni tecnologiche già esistenti e lo sviluppo di componenti vincolanti per il committente.
- Riutilizzo di componenti software OS: diminuisce la complessità generale del sistema, la durata del progetto, la difettosità del sistema.
- Uso di Software Farm OS: diminuisce i costi dei beni strumentali finalizzati alla realizzazione del sistema; riduce la dipendenza dal fornitore.
- Ricorso alle pratiche di sviluppo tipiche dei progetti OS: un ambiente di sviluppo collaborativo (tipo Sourceforge) rende più trasparente tutto il processo di sviluppo, facilita il contributo da parte di soggetti non contrattualmente rilevanti, ma interessati all'oggetto dello sviluppo; facilita la pratica del riuso applicativo.

Best practice per la valutazione delle soluzioni ad hoc, che prendono spunto dalle considerazioni svolte, potranno essere successivamente sviluppate dal Centro di Competenza e messe a disposizione delle amministrazioni.

### 3. Il Centro di Competenza

Come detto in precedenza, il modello OS è nuovo per l'Italia, e in particolare per le amministrazioni pubbliche, centrali e locali. Come tutte le cose di cui molto si parla e poco si è direttamente sperimentato, è talvolta soggetto a interpretazioni errate, che possono vanificare le potenziali ricadute positive (esaminate nei precedenti capitoli).

Per tali motivi sembra necessaria la costituzione di una struttura di riferimento sulla tematica, cui da questo momento ci si riferirà con il termine “Centro di Competenza”.

#### 3.1.1 Centri di Competenza in ambito UE

Nell'agosto 2003 la Commissione Europea, nell'ambito del programma IDA, ha deciso di incoraggiare la nascita di centri di competenza nazionali e regionali sul software OS, e di aiutarli a sviluppare interazioni tra loro, al fine di condividere applicazioni e best practice.

Tuttavia, anche se in molte nazioni europee vi sono gruppi di lavoro a livello governativo dedicati all'argomento open source (vedi paragrafo ), non sempre si può parlare di veri e propri centri di competenza istituzionali.

Con riferimento ai Centri più significativi, a livello di istituzioni europee è da citare innanzitutto l'*Open Source Observatory* (OSO) del programma IDA, che ha come scopo raccogliere e condividere risorse OS provenienti dalle amministrazioni europee, incoraggiare la diffusione e l'uso di best practice in Europa.

Tra le sezioni più significative del sito dell'OSO, si cita in particolare:

- la sezione dedicata a casi di studio;
- la sezione dedicata agli eventi relativi all'OS in Europa e fuori;
- la sezione “risorse”, ove si può trovare un elenco di centri e organizzazioni ;
- le news sulle attività dei governi europei ed extraeuropei;
- un documento sulle politiche OS in Europa e fuori Europa;
- un elenco di pubblicazioni e studi UE sull'argomento.

A livello dei singoli Paesi, i Centri e i gruppi di lavoro più significativi si trovano in Austria, Francia, Germania e Olanda.

##### 3.1.1.Austria

Il Chief Information Office<sup>73</sup> è un ufficio incaricato del coordinamento e del supporto per le iniziative e i progetti, a livello federale, per l'e-government, nonché del coordinamento con i progetti comunali e dei Länder. A esso fa capo un gruppo di lavoro sull'open source che ha un suo portale riportante studi e news<sup>74</sup>.

---

<sup>73</sup> <http://www.cio.gv.at/>

<sup>74</sup> <http://www.open-source.at/bmwa0/>. Si veda anche <http://www.cio.gv.at/alternativen/>

### **3.1.2.Francia**

L' ADULLACT<sup>75</sup> (Association des Développeurs et des Utilisateurs de Logiciels Libres pour les Administrations et les Collectivités Territoriales) è un'iniziativa nata da numerose città, comunità e consigli regionali. Questa associazione si è data il compito di coordinare le competenze delle amministrazioni e delle collettività territoriali nello sviluppo e nell'utilizzazione di software libero, nonché di assicurare un supporto tecnico di qualità ai suoi membri disponendo di un'equipe permanente di ingegneri del software di alto livello. Cura tra l'altro un archivio di software open source e una piattaforma di sviluppo collaborativo.

### **3.1.3.Germania**

Nel marzo 2004, su iniziativa del Ministero degli Interni (BMI), quello del Lavoro (BMVBW) e l'Agenzia per la sicurezza informatica (BSI) è stato costituito presso il KBSt, un centro di competenza sull'open source: OSS Kompetenzzentrum<sup>76</sup>, che raccoglie studi ed esperienze nell'ambito delle amministrazioni tedesche e ne cura il coordinamento.

### **3.1.4.Olanda**

L'Institute for Information and Communication Technology (ICTU) ha intrapreso un programma per lo "Open Standards and Open Source Software (OSOSS)" in ambito governativo, allo scopo di stabilire un catalogo di open standard raccomandati, l'introduzione ampia del modello di licenza open source in ambito governativo e l'implementazione di una piattaforma web based per l'interscambio di software tra le amministrazioni pubbliche olandesi.

Il programma<sup>77</sup> ha durata triennale e conta su di un budget di 3 milioni di euro.

## **3.2.Obiettivi del Centro di Competenza**

Il Centro di Competenza avrà come obiettivi i punti in seguito elencati:

- Sulla base delle esperienze in campo OS realizzate finora nella pubblica amministrazione, individuazione di best practice da raccomandare e diffondere, anche attraverso pubblicazioni e convegni.
- Attraverso la promozione di opportune iniziative, promozione dello sviluppo, della diffusione e della conoscenza del modello OS nell'ambito della pubblica amministrazione. Nello specifico si usa il termine "modello OS" e non semplicemente "software OS", in quanto si ritiene che il Centro di Competenza non debba limitarsi alla promozione dell'uso di software licenziato in modalità open source, ma più in generale debba informare e promuovere l'utilizzo del modello di sviluppo OS nella pubblica amministrazione.
- Raccordo dell'ampio patrimonio di esperienze svolte, in campo OS, presso l'università e la ricerca pubblica, con quello presente presso le realtà industriali del settore, in special modo tra le Piccole e Medie Imprese (PMI), anche utilizzando specifiche iniziative di

<sup>75</sup> <http://www.adullact.org/>

<sup>76</sup> <http://www.kbst.bund.de/-,247/OSS-Kompetenzzentrum.htm>

<sup>77</sup> <http://www.ososs.nl/index.jsp?alias=english>

collaborazione, convenzione e partnership. Nel perseguire questo obiettivo, il Centro favorirà l'aggregazione (in consorzi o altre forme) di PMI attive nel settore dello sviluppo e dell'integrazione di soluzioni OS e sosterrà la realizzazione di prodotti derivati e la produzione di conoscenza aggiuntiva.

- Miglioramento, attraverso proposizione di standard e metodologie, nonché tramite azioni di verifica, della qualità dei prodotti software e dei servizi di integrazione e di supporto di soluzioni basate su componenti open source offerti alle amministrazioni.
- Incontro fra domanda e offerta di prodotti e servizi OS per la pubblica amministrazione, anche tramite il portale descritto in seguito, che il Centro di Competenza dovrà realizzare nei contenuti e gestire.
- Supporto alle amministrazioni, dietro richiesta, per quanto concerne la definizione di politiche di licenziamento e di gestione dei contratti relativi a prodotti e servizi OS, adeguate per l'impiego nella PA.

Per raggiungere tali obiettivi, il Centro di Competenza dovrà prendere in carico le seguenti attività operative:

- a) Gestione del portale;
- b) Creazione e gestione della vetrina dei prodotti e servizi (sezione del portale);
- c) Raccolta moduli descrittivi di esperienze in corso nella PA (tramite il portale) e inserimento in archivio;
- d) Valutazione qualità e test di prodotti OS;
- e) Supporto e consulenza sulla redazione di Studi di Fattibilità e Capitolati Tecnici, nonché sul processo di comparazione tra soluzioni OS e proprietarie;
- f) Supporto e consulenza sulle licenze e normativa;
- g) Gestione della piattaforma di sviluppo collaborativo;
- h) Gestione relazioni nazionali e internazionali;
- i) Gestione e/o partecipazione ad attività di formazione;
- j) Gestione rapporti con il mercato e monitoraggio dello stesso;
- k) Diffusione di report sulle attività del Centro.

### ***3.3. Organizzazione del Centro di Competenza***

Il Centro di Competenza è una struttura essenziale e deve perciò disporre di una struttura permanente e dedicata, alla stregua di quanto già realizzato in numerosi paesi membri della UE.

Si ritiene che l'organizzazione del Centro debba in qualche modo riflettere i principi che sono alla base del modello OS, quindi favorire lo spirito collaborativo e lo scambio della conoscenza: il Centro deve preferibilmente configurarsi non solo come distributore, ma

soprattutto come punto di riferimento e collettore di contributi della comunità.

## 4. Portale e Vetrina

Al fine di facilitare l'incontro tra la domanda di soluzioni tecnologiche delle pubbliche amministrazioni, centrali e locali, e l'offerta di prodotti e servizi in ambito OS, il Centro di Competenza allestirà e gestirà una “vetrina virtuale”, accessibile attraverso il portale del CNIPA. Tale Vetrina ospita contenuti informativi opportuni a perseguire i seguenti obiettivi:

- introdurre le principali tematiche di riferimento nell'adozione del modello OS;
- descrivere gli attori in gioco e suggerire un modello di interazione tra questi;
- supportare le pubbliche amministrazioni nell'adozione di soluzioni idonee a ogni specifico contesto applicativo.

La Vetrina è un luogo di incontro virtuale, e quindi anche un momento aggregante per gruppi di interesse, elementi della comunità degli sviluppatori, degli integratori e degli utilizzatori.

La Vetrina, inoltre, sarà il principale strumento di disseminazione dei risultati raggiunti dal Centro di Competenza, e in prospettiva uno strumento di riferimento, per la PAC/PAL, per orientarsi in modo semplice e diretto verso i prodotti e i servizi che il mercato OS propone.

### 4.1. Modello Logico della Vetrina.

Come detto, la Vetrina costituirà il luogo virtuale per la pubblicazione delle informazioni relative alle principali risorse che possono concorrere al processo di sviluppo di sistemi software, basati anche su prodotti OS, per la pubblica amministrazione.

È utile descrivere il modello logico della Vetrina utilizzando gli strumenti dell'analisi delle basi di dati. Le entità che compongono lo schema Entità-Relazioni della Vetrina sono rappresentate in figura 4.1. Ogni entità è descritta in dettaglio nel seguito.

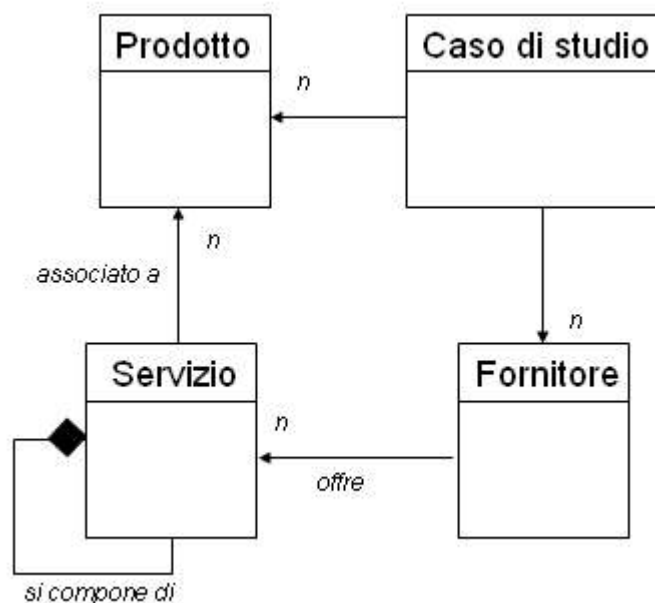


Figura 4.1: schema E-R della Vetrina



- Prodotto. La Vetrina fornisce un elenco di prodotti OS individuati dal Centro di Competenza in base alle loro caratteristiche di adottabilità nell'ambito PAC/PAL. Gli attributi obbligatori dell'entità "prodotto" sono: nome (chiave primaria), tipologia (attributo di classificazione), descrizione, sito/siti web di riferimento, attuale versione disponibile, livello di maturità determinato dal Centro di Competenza. Sono previsti ulteriori attributi facoltativi di tipo indicatori numerici quali: livello di attività del progetto, frequenza dei download del codice, numero di bug rilevati e risolti. Il complesso di questi attributi fornisce al potenziale utilizzatore una visione sintetica delle caratteristiche qualitative del prodotto software.
- Servizio. La Vetrina contiene le informazioni sui servizi associati ai singoli prodotti e a particolari integrazioni di essi che presentino una particolare rilevanza per le pubbliche amministrazioni. La gamma di servizi censiti dovrà tendenzialmente coprire tutte le possibili necessità operative, e quindi comprendere sviluppo, integrazione, formazione, supporto, manutenzione.
- Fornitore. La Vetrina ospita una sezione dedicata alle aziende che offrono servizi in ambito OS. Il Centro svolge al proposito solo una funzione di redazione e pubblicazione di informazioni, valutando esclusivamente l'applicabilità delle medesime, non formulando in nessun caso giudizi di merito o di preferenza verso singoli fornitori. Gli attributi dell'entità sono: nome (chiave primaria), tipologia di azienda (attributo di classificazione), descrizione dell'attività, servizi offerti, area geografica, riferimenti. Eventuali attributi facoltativi sono indicatori quali posizionamento sul mercato, fatturato, numero di dipendenti, ecc.
- Caso di studio. La Vetrina contiene documentazione relativa a progetti, realizzazioni, esperienze, in generale casi di studio di particolare interesse per le pubbliche amministrazioni, riguardanti realizzazioni, integrazioni e utilizzo di sistemi software basati anche su componenti OS. Le informazioni sui casi di studio sono generalmente di tipo non strutturato (documenti tecnici, interviste, valutazioni, schemi). Tuttavia, si possono individuare i seguenti attributi obbligatori dell'entità "caso di studio": nome o identificativo (chiave primaria), contesto applicativo (attributo di classificazione) amministrazioni coinvolte, fornitori coinvolti, prodotti software utilizzati, riferimenti.

## **4.2. Disegno Funzionale.**

Le risorse ospitate nella Vetrina, di tipo documentale, immagini, file multimediali, sono mantenute e gestite tramite strumenti di Content e Knowledge Management, in grado di offrire meccanismi avanzati per le seguenti funzionalità "interne":

- gestione dei contenuti (workflow per la creazione, l'approvazione e la pubblicazione);
- catalogazione, archiviazione, indicizzazione, "versioning" dei contenuti;
- gestione dei metadati;
- gestione dei modelli e delle schermate grafiche;
- gestione dati dinamici (viste su basi dati di piattaforme esterne);
- import/export;

- editor WYSIWYG<sup>78</sup>;
- gestione delle utenze (profilazione, livelli di autorizzazione alla lettura e scrittura dei contenuti, personalizzazione della presentazione sulla base del profilo utente);

Dal punto di vista degli utenti esterni, invece, le funzionalità messe a disposizione sono:

- navigazione tra i documenti, tramite opportuni link gestiti dinamicamente dallo strumento di gestione dei contenuti (ad esempio, un documento di descrizione di un prodotto dovrà offrire link a eventuali documenti su casi di studio in cui tale prodotto è stato utilizzato);
- motore di ricerca sia su testo libero sia sulla base di chiavi di classificazione e di metadati;
- forum, mailing list, sondaggi d'opinione, segnalazioni di notizie d'interesse (anche tramite newsletter).

Una rappresentazione grafica delle funzionalità offerte è mostrata in figura 4.2, nella forma di un diagramma di casi d'uso (notazione UML).

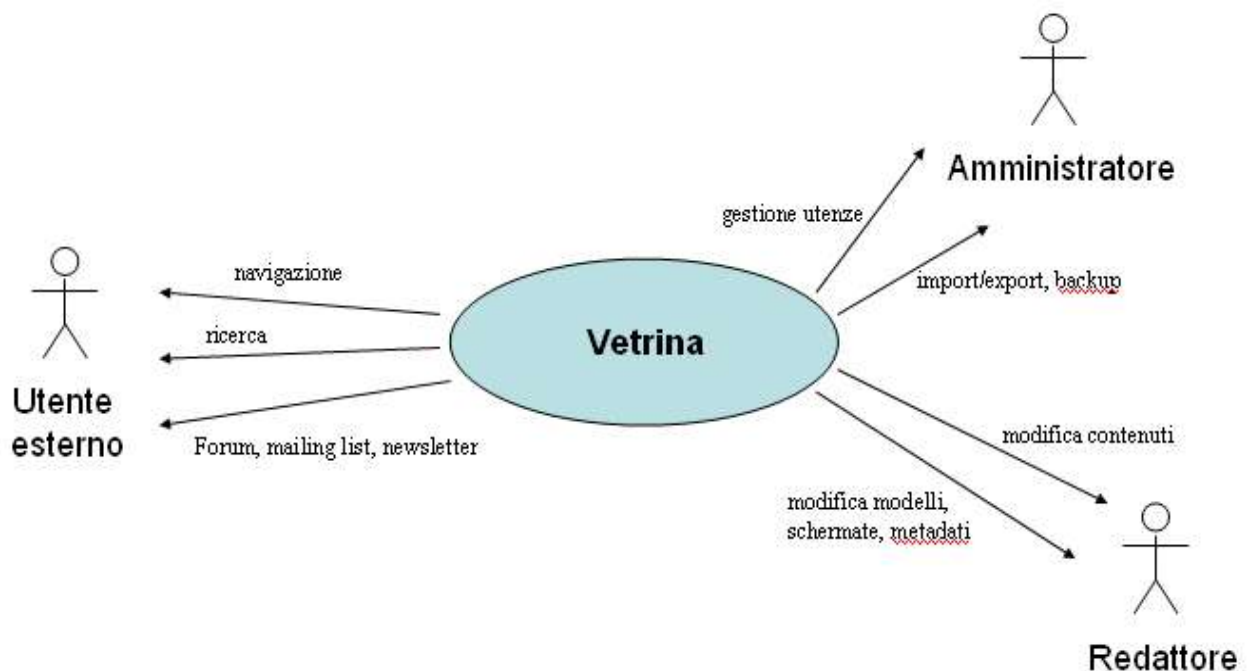


Figura 4.2: diagramma dei casi d'uso della Vetrina

Si evidenziano tre tipologie di utenti: utente esterno (fruitore del sito web), utente redattore (responsabile dei contenuti informativi e dell'aspetto grafico del sito) e utente amministratore (incaricato della gestione sistemistica del sito, compresa la gestione degli utenti registrati)

### 4.3.Mappa del sito

Nella figura seguente viene illustrato un possibile aspetto grafico della pagina iniziale della Vetrina, ove si possono identificare le varie funzionalità offerte agli utenti interni e le sezioni informativo/tematiche di cui è composta la vetrina.

<sup>78</sup> Acronimo per What You See Is What You Get

Sulla sinistra della pagina è presente il “menù di navigazione”, una serie di puntatori alle varie sezioni del sito (sezione prodotti, sezione fornitori, sezione casi di studio, ecc.).

Sulla destra è presente un riquadro con le più recenti notizie d’interesse (annunci, novità, eventi, attività in corso) e il puntatore alla form per l’accesso degli utenti registrati e l’accredito dei nuovi utenti.

La parte centrale della pagina costituisce l’area contenuti vera e propria, ove generalmente viene mostrato il documento o l’informazione multimediale che l’utente ha richiesto.

In basso a sinistra si trova il puntatore per l’accesso ai servizi riservati del sito, quali ad esempio i forum di discussione, l’area di download, i questionari di valutazione, i servizi personalizzati, ecc.

Infine, tramite i puntatori presenti sulla barra nella parte inferiore della pagina, si può accedere ai rapporti e alle statistiche sull’attività del Centro di Competenza OS del CNIPA (vedi capitolo ), ai contatti con la redazione della Vetrina, e alla mappa del sito.



Figura 4.3: pagina iniziale della Vetrina

Nelle successive figure (figura 4.4 e figura 4.5) vengono mostrate ulteriori possibili schermate del sito della Vetrina, rispettivamente la scheda descrittiva di un prodotto e di un caso di studio.

File Modifica Visualizza Preferiti Strumenti ?

Indietro Cerca Preferiti Multimedia

Indirizzo <http://www.cnipa.it/opensource/vetrina/> Val

## La Vetrina dell'Open Source nella Pubblica Amministrazione

**PRODOTTI**

- Elenco completo
- Naviga per tipologia
- Servizi associati
- Ricerca

**CASI DI STUDIO**

- Elenco completo
- Contesto applicativo
- Per amministrazioni
- Ricerca

**FORNITORI**

- Elenco completo
- Per tipologia
- Area geografica
- Ricerca

**AREA RISERVATA**

- Forum di discussione

**SCHEDA PRODOTTO PostgreSQL**

Tipologia	DBMS
Versione disponibile	7.2
Livello di maturità	3
Descrizione	PostgreSQL è un progetto open source, nato nel 1996. E' un database realizzato secondo le specifiche SQL 92 e, quindi, fortemente orientato alle transazioni. Ha un supporto all'indicizzazione sotto forma di trigger e un SQL nativo, oltre che funzioni complete per la gestione di join tra tabelle.
Valutazioni	Con la versione 7.2, PostgreSQL si conferma un database molto robusto e bene implementato, adatto alla gestione di grossi volumi di transazioni. La completa interfacciabilità e una migliorata logica di pulizia della base dati lo rendono decisamente consigliabile anche in contesti enterprise.
Siti di riferimento	<a href="http://www.it.postgresql.org">www.it.postgresql.org</a>
Fornitori in Italia	...
Servizi disponibili per le PA	...
Casi d'uso d'interesse	...

**IN EVIDENZA**

Ultime news

**RICHIESTA LOGIN**

**accesso**

se non sei ancora registrato: **registrazione**

**MINISTRO PER L'INNOVAZIONE E LE TECNOLOGIE**

**CNIPA**  
Centro Nazionale per l'Informatica nella Pubblica Amministrazione

Figura 4.4: scheda di un prodotto

File Modifica Visualizza Preferiti Strumenti ?

Indietro Cerca Preferiti Multimedia

Indirizzo <http://www.cnipa.it/opensource/vetrina/> Val

## La Vetrina dell'Open Source nella Pubblica Amministrazione

**PRODOTTI**

- Elenco completo
- Naviga per tipologia
- Servizi associati
- Ricerca

**CASI DI STUDIO**

- Elenco completo
- Contesto applicativo
- Per amministrazioni
- Ricerca

**FORNITORI**

- Elenco completo
- Per tipologia
- Area geografica
- Ricerca

**AREA RISERVATA**

- Forum di discussione

**SCHEDA CASO DI STUDIO GACC**

Tipologia	Progetto di sviluppo applicativo
Amministrazione	Scuola Superiore della P.A.
Anno di inizio del progetto	2002
Status attuale	Completato
Descrizione	E' stata realizzata la procedura di gestione amministrativa e contabile dei corsi. Consolidato un ambiente parallelo costituito da piattaforme di tipo open source e soluzioni applicative ed infrastrutturali che privilegiano le strutture informative web based. Il sito Internet è realizzato in PHP, usa il database MySQL, ed è ospitato su server Linux. Anche il nuovo sito Intranet usa la medesima piattaforma. La razionalizzazione delle procedure gestionali per la realizzazione del portale interno delle procedure, ha adottato il medesimo ambiente e analoghe modalità di sviluppo.
Valutazioni	I problemi riscontrati sono stati ... Le opportunità che hanno spinto l'iniziativa sono state:
Prodotti OS utilizzati	Linux, MySQL, PHP.
Fornitori coinvolti	...
Documentazione disponibile	...

**IN EVIDENZA**

Ultime news

**RICHIESTA LOGIN**

**accesso**

se non sei ancora registrato: **registrazione**

**MINISTRO PER L'INNOVAZIONE E LE TECNOLOGIE**

**CNIPA**  
Centro Nazionale per l'Informatica nella Pubblica Amministrazione

Figura 4.5: scheda di un caso di studio

## **5. La piattaforma di sviluppo collaborativo**

La piattaforma di sviluppo collaborativo (denominata in seguito come CDE, Collaborative Development Environment) è un ambiente tecnologico che mette a disposizione le risorse tecniche necessarie alla creazione di un eco-sistema, a iniziativa nazionale, per lo sviluppo di sistemi OS.

### **5.1. Modelli e strumenti per lo sviluppo collaborativo**

Il modello di riferimento per la definizione della CDE è quello ben noto di SourceForge. Quest'ultimo è un portale che offre servizi gratuiti agli sviluppatori di software OS. Attualmente il portale ospita oltre 80.000 progetti e ha un numero di sviluppatori iscritti che si avvicina alle 900.000 unità. Le ragioni di questo enorme successo sono da ricercarsi nella notevole semplicità d'uso dell'interfaccia, nella potenza degli strumenti messi a disposizione, nell'organizzazione del sito, che fornisce un micro-portale dedicato a ciascun progetto. L'insieme di queste caratteristiche, unite al fatto che queste iniziative creano un circolo virtuoso (più sviluppatori sono iscritti, più altri sviluppatori sono motivati a iscriversi), permettono a SourceForge di essere il punto di riferimento per lo sviluppo collaborativo.

Analogamente, la piattaforma CDE ha lo scopo di fornire tutti gli strumenti abilitanti destinati alla creazione di componenti software OS. Gli attori che interagiranno in questo contesto saranno:

- sviluppatori di software appartenenti a PMI,
- sviluppatori indipendenti,
- realtà industriali di grandi dimensioni,
- istituti di ricerca e formazione.

Il Centro di Competenze del CNIPA sarà protagonista dell'iniziativa, non solo come fornitore e gestore del CDE, ma anche e soprattutto come ispiratore delle iniziative progettuali, potendo avvalersi di un osservatorio privilegiato delle necessità informatiche delle singole amministrazioni.

### **5.2. Requisiti funzionali del CDE**

Le funzionalità messe a disposizione dal CDE possono essere modulate in funzione della crescita quantitativa e qualitativa della comunità che andrà a convergere intorno ad esso. È prevedibile che in una fase iniziale siano erogate funzionalità di base, volte a divulgare alcune iniziative di sviluppo prioritarie, insieme agli strumenti per la gestione dei progetti. In una seconda fase, si potranno aggiungere nuove funzionalità, più sofisticate, volte a migliorare l'interazione tra gli attori dell'ecosistema.

Si possono elencare un insieme indicativo di funzionalità e strumenti, il cui rilascio sarà programmato nel tempo in sede di progetto esecutivo:

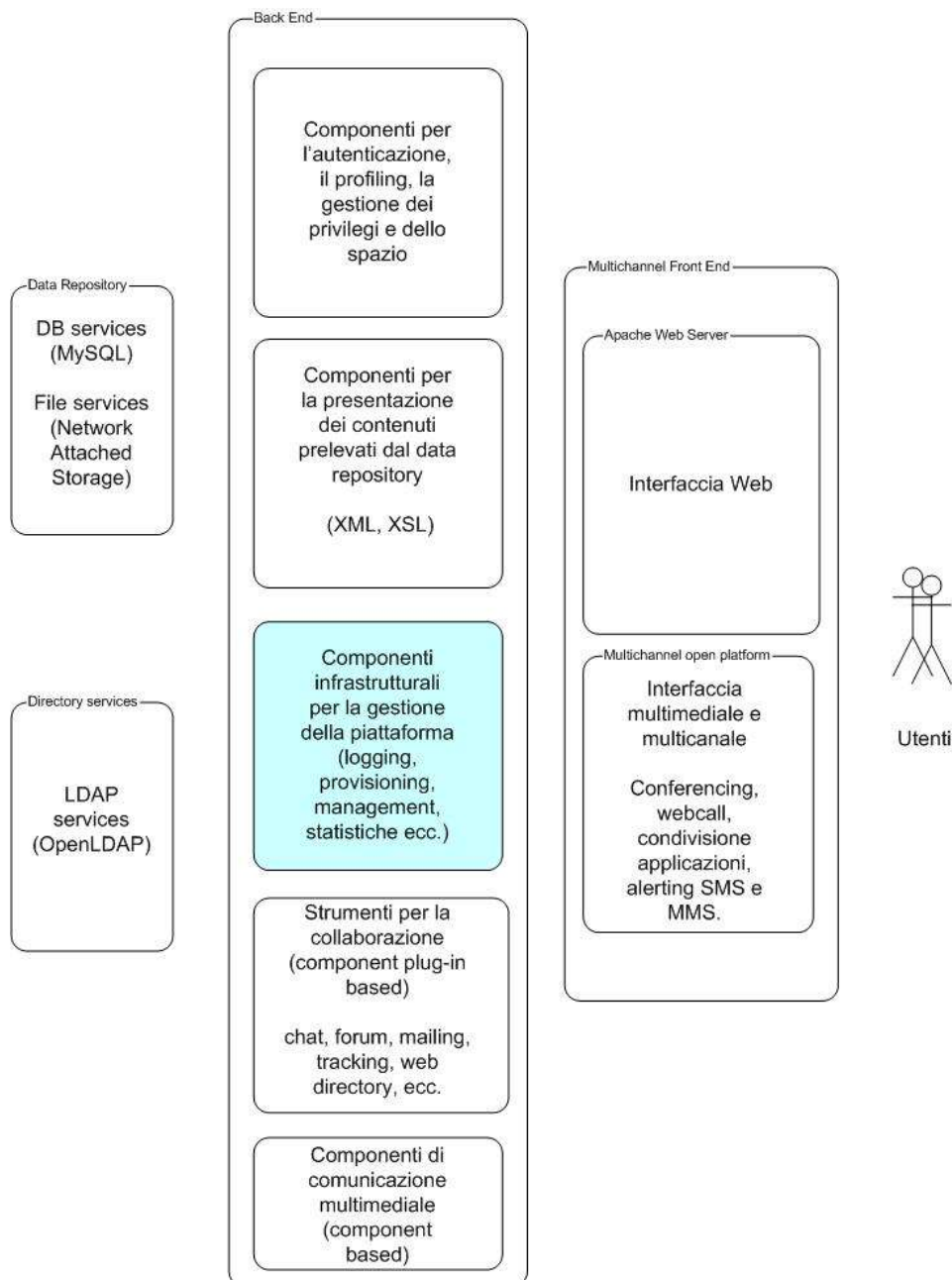
- strumenti per la gestione e amministrazione dei progetti (strumenti di collaboration, workflow, forum, news, modelli per schede descrittive e di sintesi, generazione automatica di schede identificative delle componenti);

- tracking delle versioni e dei bug;
- strumenti per la generazione automatica della documentazione;
- mailing list e area contatti;
- catalogo di progetti navigabile;
- strumenti per la gestione della configurazione (CVS);
- download di file (sorgenti, patch, documentazione, ecc.) ;
- strumenti per la formazione basata su contenuti multimediali (webcasting);
- strumenti per la collaborazione avanzata in rete (webconferencing);
- motore di ricerca;
- gestione profilo utenti;

### ***5.3.Architettura.***

Lo schema funzionale della piattaforma CDE è rappresentato in figura 5.1.





**Figura 5.1: schema funzionale della piattaforma CDE**

Nella figura si nota la suddivisione classica in tre livelli (3-tier architecture): gli utenti interagiscono con il sottosistema di front-end che si occupa dell'esposizione delle interfacce (tier 1), il front-end a sua volta comunica con il back-end (tier 2), il quale implementa l'intera logica applicativa della piattaforma utilizzando, fra l'altro l'archivio dati (tier 3), composto dal DB, dal file system di rete e dai directory services.

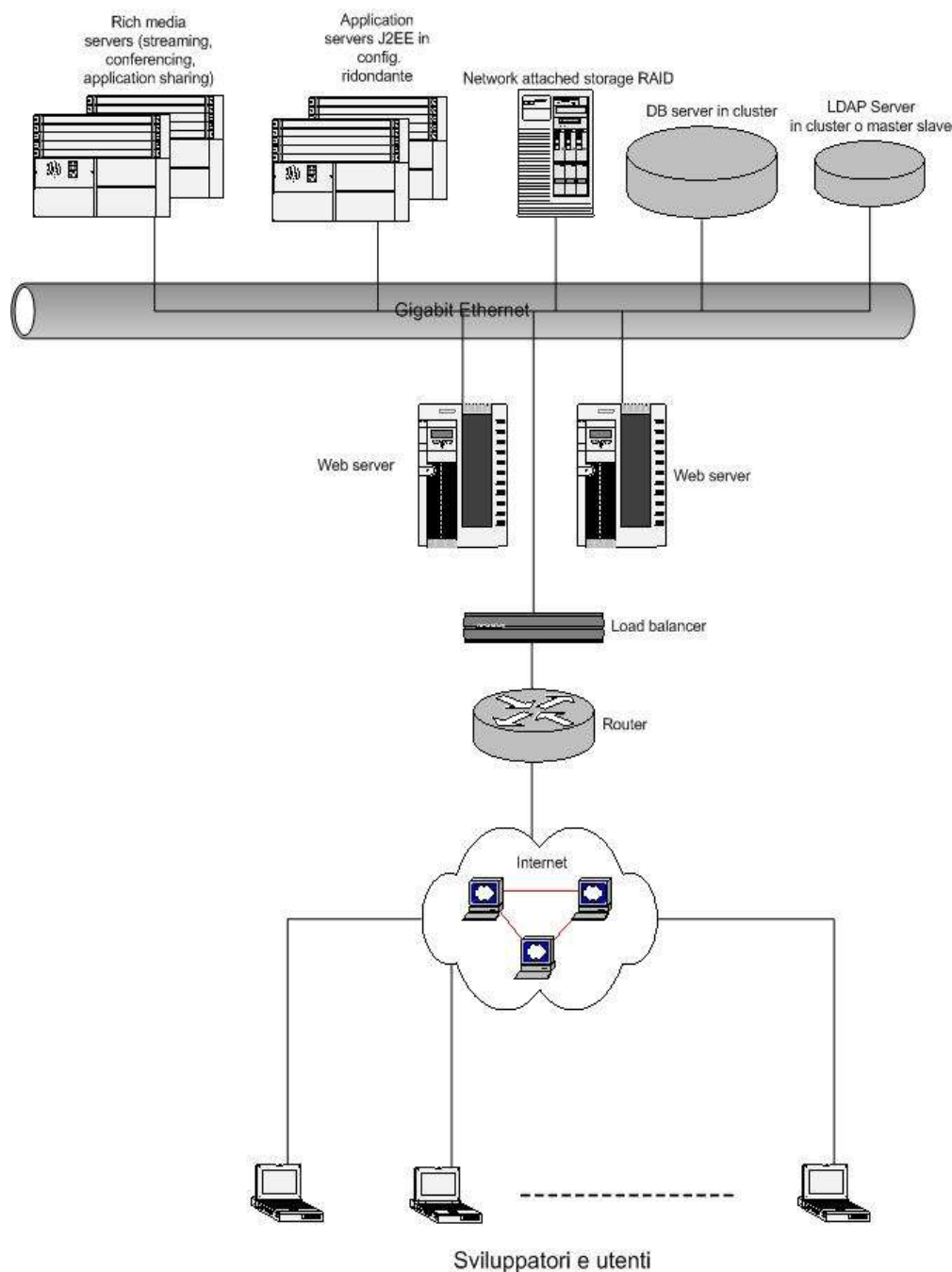
La logica di piattaforma risiede interamente nella parte di back-end, e viene implementata su piattaforma J2EE. Sono state isolate le principali componenti funzionali, rese modulari per consentire una corretta gestione della complessità, inoltre, la modularità degli elementi di logica applicativa è necessaria per la scalabilità e l'affidabilità dell'intero sistema.

I sottosistemi funzionali di back end raggruppano le funzionalità di: autenticazione, profilo utenti, gestione dei privilegi, schemi di visualizzazione personalizzati in base al profilo utente, logica applicativa degli strumenti di collaborazione (tracking, CVS, chat, forum, ecc.), connettività verso i server di videoconferenza, sistema di erogazione di flussi multimediali in

Video On Demand, sistema per la collaborazione applicativa, ecc. Inoltre, il back end raggruppa tutte le funzioni di base della piattaforma (gestione, provisioning, logging, statistiche, ecc.).

Per la sezione di DB e directory services sono stati selezionati i due prodotti open source più stabili e diffusi, rispettivamente MySQL e OpenLDAP. Questi strumenti offrono caratteristiche di affidabilità adeguate al tipo di progetto.

Uno schema di massima della architettura hardware è rappresentato in figura 5.2. Gli elementi presenti nello schema permettono di erogare tutte le funzionalità proposte. L'affidabilità e la scalabilità dell'architettura vengono garantite da meccanismi di ridondanza e clustering, tipici in questo genere di sistemi.



**Figura 5.2: architettura**



#### **5.4.Strumenti tecnici innovativi.**

Alcuni aspetti del modello di sviluppo OS possono essere meglio supportati con l'introduzione di strumenti avanzati che incrementino l'efficienza e la qualità complessiva dei processi in generale e dell'interazione tra gli attori dell'ecosistema in particolare.

Gli strumenti avanzati, che prospetticamente consideriamo nell'ambito del CDE, sono tutti quelli in grado di mettere in contatto gli elementi della comunità attraverso canali che privilegiano il fattore multimediale. Il contatto diretto tra gli sviluppatori avviene, quindi, anche attraverso le immagini, la voce, la condivisione real time delle applicazioni. I tutorial non sono più solo dei manuali, ma si evolvono e diventano presentazioni multimediali, nelle quali la sequenza di slide viene commentata a voce e lo speaker può apparire in video. Il lancio di una nuova versione di un software OS può essere commentato dal gruppo di sviluppatori in un evento di live webcasting.

La comunità di utenti, per poter seguire il tutto, ha semplicemente bisogno di un browser web e di alcuni client OS specifici, oltre che, naturalmente, di una buona connettività. In un certo senso, grazie agli strumenti evoluti, viene potenziato il fattore umano, la conoscenza personale. Questi elementi portano a una collaborazione ancor più efficace.

L'unico lato negativo di questi strumenti, come si accennava, è nell'occupazione di risorse. Tuttavia, un'attenta razionalizzazione e una buona gestione della scalabilità permettono l'introduzione di queste caratteristiche evolute in modo graduale, proporzionale all'effettiva richiesta da parte degli utilizzatori.