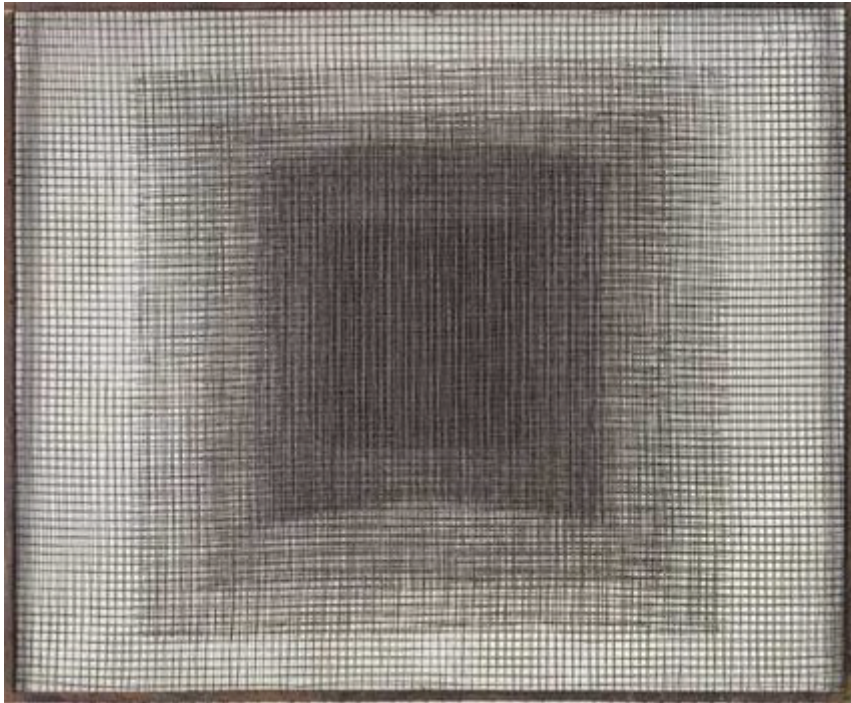


Andrea Rosada

Architettura Open Source



Tesi di Dottorato in Architettura e Progettazione Edilizia
Politecnico di Torino
XXVI Ciclo (2011-2013)

Andrea Rosada

Architettura Open Source

Tesi di Dottorato in Architettura e Progettazione Edilizia
Politecnico di Torino
XXVI Ciclo (2011-2013)

Architettura Open Source

Tesi di Dottorato in Architettura e Progettazione Edilizia - DAPe
XXVI Ciclo (2011-2013)

DAD - Dipartimento di Architettura e Design
Politecnico di Torino
Viale Mattioli 39
10125, Torino, Italy

Candidato: Andrea Rosada
Matricola: 178645

Tutor: Pierre-Alain Croset
Coordinatori DAPe: Antonio De Rossi - Giovanni Durbiano

Immagine in copertina: Francesco Lo Savio, Filtro, 1962
Tele sovrapposte, 100 x 120 cm

Indice

Ringraziamenti	i
Introduzione	1
I Mondo Open	5
1 Open Source	9
1.1 Internet e WWW	9
1.1.1 Internet	9
1.1.2 Il funzionamento della rete	11
1.1.3 Il Web	13
1.1.4 Il personal computer	14
1.1.5 Hackers	16
1.2 Dal free software all'Open Source	18
1.2.1 Free software	18
1.2.2 Open Source	21
1.2.3 Anarchia organizzata	24
1.2.4 Reputazione e collaborazione	26
1.2.5 Modularità e ricombinazione	28
2 Open Design	31
2.1 Dal bit all'atomo	31
2.1.1 Non solo software	31
2.1.2 Dal bit all'atomo	33
2.1.3 Fab	34
2.1.4 Open Design	37
2.1.5 Casi esemplari di Open Design e Open Hardware	41
2.1.6 Arduino	45
2.1.7 Il processo dell'Open Design	47
2.1.8 Open-X	50

II	Architettura Open Source	53
3	Architettura aperta	57
3.1	Processo e utente	57
3.1.1	Opera aperta	57
3.1.2	La fine dell'architetto demiurgo	59
3.1.3	L'utente all'interno del processo edilizio	61
3.1.4	Strumenti di coinvolgimento	62
3.2	Architettura aperta	66
3.2.1	Alexander e il 'pattern language'	67
3.2.2	Habraken e i 'supports'	72
3.2.3	Friedman e il 'flatwriter'	78
3.2.4	Il metodo Segal	81
4	Architettura Open Source	85
4.1	Il dibattito contemporaneo	85
4.1.1	Primi contributi all'Architettura Open Source	85
4.1.2	Il dibattito attuale	91
4.1.3	Una definizione ancora aperta	95
4.2	Lettura di un fenomeno emergente	96
4.2.1	Iniziative di Architettura Open Source	96
4.2.2	Un corpus di esperienze eterogeneo	97
4.3	Casi studio	99
4.3.1	Open Architecture network	99
4.3.2	Open Structures	105
4.3.3	OpenSimSim	109
4.3.4	Air Tree Commons	113
4.3.5	Open Source Ecology	117
4.3.6	WikiHouse	122
4.4	Considerazioni generali	128
4.4.1	Temi e campi di applicazione	128
4.4.2	L'Architettura Open Source come strumento	129
4.4.3	Oltre l'emergenza e i Paesi in via di sviluppo?	130
4.4.4	Possibilità effettive di applicazione	132
III	Strumenti operativi dell'Architettura Open Source	135
5	Strumenti operativi	139
5.1	Il progetto della sorgente	139

5.1.1	Sorgente uguale apertura	140
5.1.2	Ricombinazione e modularità	141
5.1.3	Dal bit all'atomo	142
5.2	Il progetto di gestione della comunità	143
5.2.1	Figure e ruoli	144
5.2.2	Auto-organizzazione e autorialità	145
5.2.3	Organizzazione modulare della comunità	147
5.3	Il progetto della piattaforma	148
5.3.1	Strumenti e pratiche	149
5.3.2	Oltre il repository	150
5.3.3	Piattaforma e interazione	151
6	Applicazione pratica	153
6.1	BrickShell	154
6.1.1	Obiettivi, ipotesi iniziali e adozione dell'Open Source	157
6.1.2	Tema e campo di applicazione	158
6.1.3	Definizione della sorgente	158
6.1.4	Definizione della comunità	160
6.1.5	Definizione della piattaforma	161
6.1.6	Svolgimento	163
6.2	Utilità della sperimentazione	173
6.2.1	BrickShell: Architettura Open Source	173
6.2.2	BrickShell: sorgente progettuale	176
6.2.3	BrickShell: comunità di pari	178
6.2.4	BrickShell: piattaforme di supporto	180
6.2.5	BrickShell: considerazioni generali e risultati	181
	Conclusioni e sviluppi futuri	185
	Transcalarità	185
	Un nuovo ruolo per l'architetto?	186
	Il dibattito continua	186
	Open Source e pratica professionale	188
	Conclusioni	189

Ringraziamenti

Una tesi sull'open source non può che basarsi su tanti contributi diversi e configurarsi come un continuo processo di mutuo apprendimento. Per questo motivo devo ringraziare le persone che ho avuto la possibilità di incontrare nel mio percorso di ricerca.

Pierre-Alain Croset ha accolto l'idea della tesi con entusiasmo e curiosità, fattori che, oltre ai consigli preziosi e alle attente disamine scientifiche e disciplinari, hanno contribuito enormemente allo sviluppo del mio lavoro. È vero che la ricerca scientifica è questione di metodo, ma è ancor più vero che è questione di atteggiamento e spirito critico: vederlo all'opera con gli studenti, i colleghi e con me è stato (ed è) un continuo motivo di stimolo intellettuale.

Un importante ruolo, prima ancora che iniziassi a lavorare in Dipartimento, lo hanno avuto Stefano Pujatti, Valeria Brero, Corrado Curti e Daniele Almondo e tutta l'allegria banda di cascina Giardina. Aver avuto la possibilità di lavorare e confrontarmi con loro è per me motivo di grande orgoglio e crescita professionale.

Mario Sassone mi ha dato la possibilità di mettere a punto gli strumenti definiti durante il lavoro di ricerca attraverso l'organizzazione e la gestione del workshop BrickShell. Grazie a lui e a Tomas Mendez Echenagucia, Iasef MD Rian, Shaghayegh Rajabzadeh e a tutte le persone che vi hanno partecipato (studenti e ospiti) ho avuto modo di sperimentare nella pratica ciò che fino ad allora era solo teoria, compiendo quello che reputo un passo decisivo per l'avanzamento della mia ricerca.

Angela Lacirignola ha messo a disposizione il laboratorio LATEC, il suo tempo e le sue energie per permetterci di portare a termine la costruzione di BrickShell.

Enrico Boffa, Ilaria Ariolfo, Matteo Malandrino e Antonio Spinelli insieme con gli studenti, i docenti membri e i coordinatori del collegio DAPe sono stati, attraverso il loro lavoro e le loro critiche, una fonte inesauribile di spunti e riflessioni.

Alastair Parvin, Tatjana Schneider, Francesco Cingolani e Daniel Dendra, seppur in brevi incontri, hanno contribuito ad approfondire la mia conoscenza su molti dei casi che ho preso in esame.

JENGA! e /LAM sono stati spazi di confronto e maturazione che, anche se lontani dai temi di ricerca, hanno avuto e continuano ad avere un ruolo importante per quanto riguarda l'attività professionale e la mia visione dell'architettura.

Infine debbo ringraziare la mia famiglia e soprattutto Laura, il cui supporto quotidiano è stato fondamentale.

Introduzione

“Sono giunto al termine di questa mia apologia del romanzo come grande rete. Qualcuno potrà obiettare che più l’opera tende alla moltiplicazione dei possibili più s’allontana da quell’unicum che è il self di chi scrive, la sincerità interiore, la scoperta della propria verità. Al contrario, rispondo, chi siamo noi, chi è ciascuno di noi se non una combinatoria d’esperienze, d’informazioni, di letture, d’immaginazioni? Ogni vita è un’enciclopedia, una biblioteca, un inventario d’oggetti, un campionario di stili, dove tutto può essere continuamente rimescolato e riordinato in tutti i modi possibili. Ma forse la risposta che mi sta più a cuore dare è un’altra: magari fosse possibile un’opera concepita al di fuori del self, un’opera che ci permettesse d’uscire dalla prospettiva limitata d’un io individuale, non solo per entrare in altri io simili al nostro, ma per far parlare ciò che non ha parola, l’uccello che si posa sulla grondaia, l’albero in primavera e l’albero in autunno, la pietra, il cemento, la plastica... Non era forse questo il punto d’arrivo cui tendeva Ovidio nel raccontare la continuità delle forme, il punto d’arrivo cui tendeva Lucrezio nell’identificarsi con la natura comune a tutte le cose?”

(Calvino 1993, p. 134-135)

Internet e le tecnologie dell’informazione e della comunicazione hanno modificato radicalmente gli ultimi decenni del XX secolo e continuano a influenzare la nostra società. Non è cambiato solo il modo di accedere e di fruire dell’informazione ma anche il modo attraverso il quale l’informazione viene prodotta. Infatti negli ultimi anni si è assistito all’emergere di importanti modelli di produzione basati sulla collettività, la collaborazione e l’organizzazione autonoma piuttosto che sulla gerarchia e sul controllo. Alcuni di questi, come l’Open Source, si sono affermati non solo nel campo della produzione di beni intangibili, ma anche nello sviluppo di beni materiali.

L’open source è emerso a partire dagli anni Novanta del XX secolo grazie al suo successo nella produzione di software affidabile e robusto. Questo paradigma di produzione di software, generatore di programmi creati in maniera collettiva seguendo un processo in cui gli utenti sono anche (in misure differenti) gli sviluppatori, ha sottolineato l’efficacia dei sistemi aperti organizzati su base comunitaria orizzontale rispetto agli standard proprietari, evidenziandone l’economicità e la flessibilità di adattamento a differenti situazioni. Nel mondo della produzione immateriale (informazioni, idee, programmi etc.) l’Open Source è riuscito a garantire alla collettività accesso immediato alla tecnologia normalmente in uso, grazie a una ridefinizione del processo di sviluppo e distribuzione ridisegnando i ruoli degli attori coinvolti. La sfida di ‘aprire’ il mondo fisico è però più complessa. Il tentativo di

capire come e se un oggetto si può definire ‘open source’ va immediatamente a toccare aspetti legati alla sua produzione sia materiale che immateriale: il materiale di cui è fatto, come è stato prodotto, e al contempo come è stato ideato, disegnato, standardizzato, ma anche aspetti legati al tipo di utilizzo che è possibile farne.

In ambito architettonico la messa in pratica di queste idee rilancia una visione già espressa dalle tecno-utopie qualche decennio fa e abbandonata a metà degli anni Settanta. Questa visione di tecnologia inclusiva si è evoluta con l’avvento del ‘computer-aided design’ prima, e con l’esplosione del web 2.0 in un secondo momento, usando la tecnologia come mezzo per incoraggiare la partecipazione degli utenti e per rendere i ‘non esperti’ in grado di esprimere direttamente i loro bisogni e desideri al di là, o addirittura senza la mediazione di l’architetto. In un sistema globale sempre più complesso e stratificato, dove i ruoli e i mandati di utenti e progettisti sono sempre meno definiti (l’immagine di copertina, un’opera di Francesco Lo Savio, rappresenta bene questa situazione di indefinitezza) è necessario riflettere sulle implicazioni che l’evoluzione tecnologica digitale ha sull’architettura e sui cambiamenti che essa comporta.

L’obiettivo di *Architettura Open Source* è verificare l’ipotesi che il fenomeno Open Source sia applicabile anche all’interno del processo edilizio, attraverso la definizione di opportuni strumenti che si basano su un uso collaborativo, aperto e partecipativo delle attuali tecnologie digitali di comunicazione.

Di seguito verrà illustrata brevemente la struttura di questa tesi e l’organizzazione dei suoi contenuti. Nella prima parte del lavoro si è cercato di dimostrare che, a partire dall’affermarsi di Internet fino al contemporaneo sviluppo delle ICT, il mondo attuale sta vivendo delle trasformazioni sostanziali, trasformazioni che possono avere dei riflessi anche nel campo dell’architettura e dei suoi metodi di sviluppo e di produzione. Più che sottolineare le vicissitudini storiche, il lavoro di ricerca intende evidenziare il fatto che Internet si sia fatta vettore di aspirazioni e bisogni latenti, favorendo la nascita e lo sviluppo di fenomeni innovativi e propositivi, alternativi o comunque estranei al trend imperante basato sulla concorrenza piuttosto che sulla collaborazione. Nel primo capitolo viene analizzato il fenomeno dell’Open Source come nuovo paradigma di produzione dell’informazione; nel secondo capitolo si vedrà come l’Open Source non si sia affermato unicamente nel campo dell’informatica e dei software, ma abbia anche influenzato la produzione di contenuti digitali fino a interessare il mondo del design e della progettazione e produzione di oggetti. Alla fine del secondo capitolo ci si soffermerà sull’Open Design e sulle sue caratteristiche di funzionamento.

All’interno dei due capitoli che compongono la seconda parte (terzo e

quarto capitolo) si ragiona sulla plausibilità della tesi e sui fenomeni emergenti (tendenzialmente recenti) che si pongono come applicazioni dell'Open Source in architettura. Viene affrontato il tema della partecipazione mediata dalla tecnologia, ovvero di quelle teorie architettoniche che hanno sviluppato strumenti appositi per permettere all'utente finale di partecipare all'intero processo edilizio. Partendo da visioni e teorie sviluppatesi a partire dagli anni Sessanta del secolo scorso e arrivando fino ai più recenti casi di applicazione dell'Open Source, si vedrà come la ricerca in architettura abbia affrontato il tema dell'apertura del processo edilizio. Nel terzo capitolo viene discusso il tema dell' 'opera aperta', trattato in campo architettonico da alcuni autori come Alexander, Habraken, Friedman e Segal. Ciascuno di questi ha messo a punto appropriati strumenti e tecniche per favorire l'inclusione e la partecipazione dell'utente finale all'interno del processo edilizio, processi e tecniche che, riletti sotto la luce del paradigma della 'network society', risultano ancora attuali ed efficaci. Nel quarto capitolo si affronta il dibattito recente sull'Architettura Open Source, valutando i diversi modi con cui è stato trattato nelle riviste di settore e dagli addetti ai lavori. Oltre a ciò verranno analizzate alcune delle iniziative recenti che si pongono l'obiettivo di applicare l'Open Source all'architettura, da cui è possibile dedurre alcune considerazioni generali.

Nella terza e ultima parte viene esposto il contributo originale della tesi. Attraverso lo studio del fenomeno Open Source e l'analisi dei casi studio nei capitoli precedenti è stato possibile definire alcuni strumenti operativi che permettono di implementare l'Open Source all'interno del processo edilizio. La necessità di testarne il funzionamento e gli effetti ha portato alla realizzazione di una esperienza di Architettura Open Source a scopi didattici all'interno dell'offerta formativa del Dipartimento di Architettura e Design del Politecnico di Torino. Tale iniziativa, denominata Brickshell, è stata il banco di prova dell'efficacia degli strumenti operativi definiti in precedenza. Nel quinto capitolo vengono elencati e illustrati gli strumenti operativi dell'Architettura Open Source, mentre nel sesto capitolo verrà chiarito l'utilizzo di tali strumenti, attraverso la descrizione del workshop BrickShell.

Infine si conclude la trattazione presentando i risultati dell'esperienza BrickShell e alcuni possibili futuri sviluppi di questa ricerca.

Parte I
Mondo Open

“Le ICT [...] sono dispositivi che comportano trasformazioni radicali, dal momento che costituiscono ambienti in cui l’utente è in grado di entrare tramite porte di accesso (possibilmente amichevoli), sperimentando una sorta di iniziazione. Non vi è un termine per indicare questa nuova forma radicale di costruzione, cosicchè possiamo usare il neologismo riontologizzare per fare riferimento al fatto che tale forma non si limita solamente a configurare, costruire o strutturare un sistema (come una società, un’auto o un artefatto) in modo nuovo, ma fundamentalmente comporta la trasformazione della sua natura intrinseca, vale a dire della sua ontologia. In tal senso, le ICT non stanno soltanto ricostruendo il nostro mondo: lo stanno riontologizzando.”

(Floridi 2012, p. 13)

I cambiamenti che avvengono nella nostra società, il paradigma tecnologico che porta alla network society, la rivoluzione digitale e la crisi economica in atto stanno portando a un radicale cambiamento dei nostri sistemi di produzione. Grazie allo sviluppo di Internet, non è solo aumentato radicalmente il numero di informazioni che il genere umano produce e a cui può accedere, la nostra modalità di comunicazione e di accesso all’informazione, ma si sono sviluppati anche altri metodi di produzione, alternativi o comunque differenti rispetto ai metodi tradizionali. Se questo enorme sistema informazionale che abbiamo creato nel corso degli anni sta cambiando il nostro modo di fruire dell’informazione, di rapportarci con la cosa pubblica e con lo stato e forse, più in generale, sta modificando il nostro stile di vita, lo stesso vento di cambiamento permea quasi tutti i campi del sapere e dell’attività umana.

In questi due primi capitoli si vuole dimostrare che grazie all’introduzione delle tecnologie informatiche, in particolare di Internet e del Web, ci si sta spostando da un sistema di innovazione che fa capo a un sistema centralizzato e verticale di informazione (intesa nel senso più ampio e vasto del termine) a un sistema orizzontale, plurale, apparentemente caotico ma efficace e produttivo allo stesso tempo, di cui il movimento Open Source diviene esempio paradigmatico.

Capitolo 1

Open Source

1.1 Internet e WWW

1.1.1 Internet¹

Nato ufficialmente il 1 gennaio 1983, Internet è una rete mondiale di computer interconnessi tra di loro (Internet infatti sta per INTERconnected NETWORKS) e assieme ad altre tecnologie (come ad esempio il GPS e la tecnologia GSM) costituisce l'infrastruttura tecnologica comunemente definita ICT, Information and Communication Technology, attraverso cui ci scambiamo e processiamo quotidianamente informazioni.

Internet è l'erede diretto di ARPANET, una rete di calcolatori interconnessi tra di loro sviluppata sotto la spinta del governo degli Stati Uniti d'America attraverso l'ARPA (Advanced Research Projects Agency) con lo scopo di collegare i calcolatori dei maggiori centri di ricerca americana per favorire lo scambio reciproco di informazioni. L'ARPA era stata fondata nel 1958 su suggerimento di James R. Killian jr., l'allora presidente del MIT (Massachusetts Institute of Technology), al presidente Eisenhower il quale, allarmato dalla messa in orbita del primo Sputnik da parte dei sovietici solo un anno prima, temeva di perdere la supremazia militare e tecnologica nei confronti dell'Unione Sovietica. Eisenhower, consigliato dal Comitato di consulenza scientifica presidenziale (Senior Advisor Committee), aveva dato

¹In questo paragrafo si fa esplicitamente riferimento, ove non indicato, al quarto capitolo del libro *Libertà di software, hardware e conoscenza*, di M. Berra e A. R. Meo, 2006 e al lavoro di Carlo Gubitosa, *La vera storia di Internet*, disponibile all'indirizzo: <http://www.apogeonline.com/2001/libri/88-503-1055-2/ebook/pdf/StoriaInternet.pdf>. La storia di Internet non è oggetto della tesi, la sua trattazione è strumentale alla redazione in quanto fornisce spunti interessanti circa i fenomeni che si sono sviluppati grazie ad essa.

pieni poteri a Killian per la costituzione di una agenzia indipendente che potesse rivaleggiare con la potenza scientifica sovietica, la quale aveva messo in campo un esercito di ricercatori (nell'ordine di un milione di studiosi) e, con grande probabilità, aveva accumulato un grande patrimonio di conoscenze scientifiche in molti campi del sapere. La convinzione poi che nel mondo della scienza, a differenza di quello delle tecnologie applicate per la guerra o per il mercato, la chiave del successo dovesse essere ricercata nella collaborazione e non nella competizione, portò allo sviluppo a partire dal 1969, nell'ambito della sperimentazione sul networking, di ARPANET, l'embrione di ciò che sarebbe poi diventato Internet. ARPANET inizialmente connetteva quattro

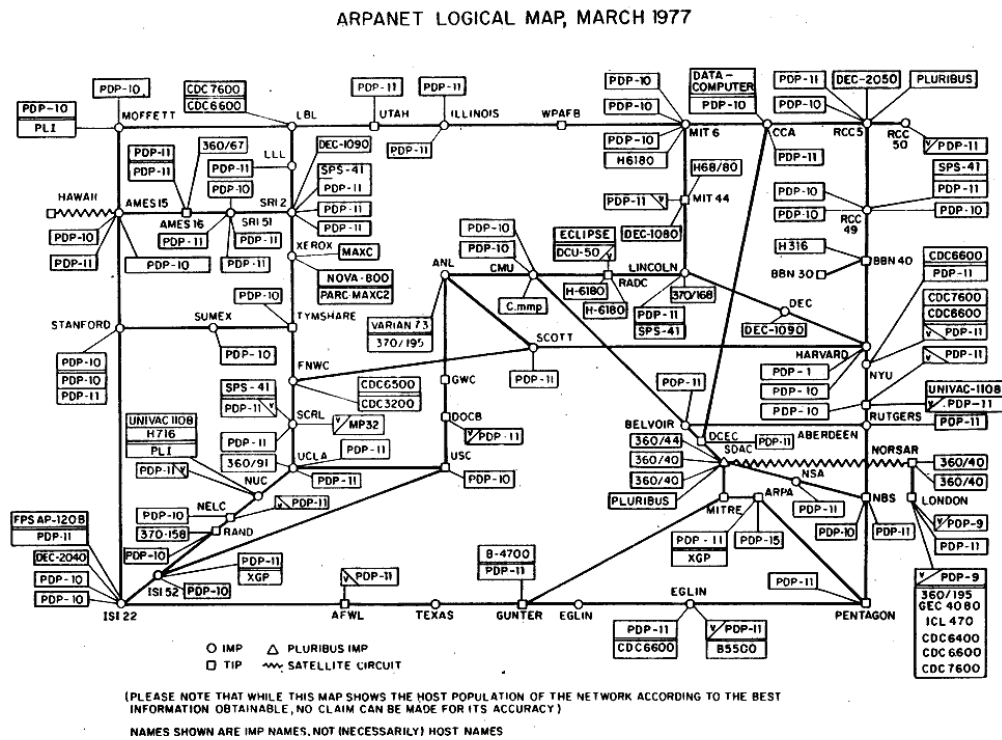


Figura 1.1: La mappa logica di ARPANET così come appariva nel marzo 1977. Si possono notare i principali nodi americani facenti capo alle università e alcuni nodi europei. L'immagine è tratta da Wikipedia, all'indirizzo <http://it.wikipedia.org/wiki/ARPANET>.

nodi corrispondenti a quattro diverse università americane: l'Università della California di Los Angeles, lo Stanford Research Institute, l'Università di California di Santa Barbara e infine l'Università dello Utah. Nel giro di pochi anni i nodi si moltiplicarono sul territorio statunitense fino a comprendere nel 1973 anche i primi nodi europei. Ad ogni nodo corrispondeva un computer (detto mainframe) o un insieme di calcolatori che risultavano essere grossi più

o meno come un armadio e riempivano intere stanze (Figura 1.1). Il personal computer così come lo conosciamo ancora non apparteneva a questi anni. La storia di Internet prende piede a partire dal 1982, quando ARPAnet si divide in MILnet (rete interna al ministero della difesa e dedicata ad applicazioni militari, poi abbandonata) e in Internet (con la I maiuscola, per differenziarla dalla sua tecnologia che porterà lo stesso nome ma con la 'i' minuscola), e dal quel momento Internet si svilupperà fino ad avere l'aspetto (che in realtà è in continua mutazione) che tutti noi conosciamo e sperimentiamo quotidianamente.

1.1.2 Il funzionamento della rete

“Tuttavia un aspetto è sempre rimasto immutato: internet non ha alcun direttorio centrale che ne guidi lo sviluppo; al contrario, la sua tecnologia viene ancora sviluppata da una comunità aperta di hacker.”
(Himanen 2001)

Internet è libero. Questa affermazione che potrebbe sembrare pura retorica si basa in realtà su solide basi pratiche: Internet non è di proprietà di nessuno (governi, imprenditori, lobby) ma è un strumento dotato di una propria autonomia. Autonomia che viene garantita da specifici organismi, quali l'Internet Engineering Task Force e l'Internet Society. Questi, al di là dei nomi che potrebbero anche risultare altisonanti, sono in realtà delle comunità aperte di cosiddetti 'hackers' che sviluppano software per il funzionamento di internet. Le due comunità sono totalmente aperte ed è possibile parteciparvi anche semplicemente iscrivendosi alla loro mailing list. A dimostrazione di quanto detto si deve pensare al protocollo TCP/IP, sviluppatosi all'interno di ARPA, e alle sue alternative, x.25 e ISO, promosse dalle due maggiori organizzazioni per la standardizzazione (CCITT e ISO). Sembra che una delle cause per cui i protocolli proposti da queste due organizzazioni non abbiano funzionato e non abbiano preso il sopravvento, sia la loro natura sostanzialmente chiusa (Abbate 1999). L'esempio del protocollo TCP/IP, che è riuscito a resistere ai tentativi di imposizione da parte di importanti organismi internazionali, dimostra come sia difficile mettere 'il cappello' a Internet e alla comunità che vi lavora e lo sviluppa incessantemente. Dal punto di vista tecnico, invece, internet funziona grazie alla commutazione di pacchetto. Quest'ultima, a differenza della commutazione di circuito che governa l'utilizzo dei telefoni, ottimizza l'impiego della rete perché permette a più stazioni la trasmissione di diversi messaggi sullo stesso canale. La struttura di ARPANET (che sarà poi anche quella di internet) è una struttura a network (Figura 1.2), ovvero una struttura simile al cervello umano, all'interno del quale "i neuroni e loro collegamenti sono molteplici e sovrabbondanti,

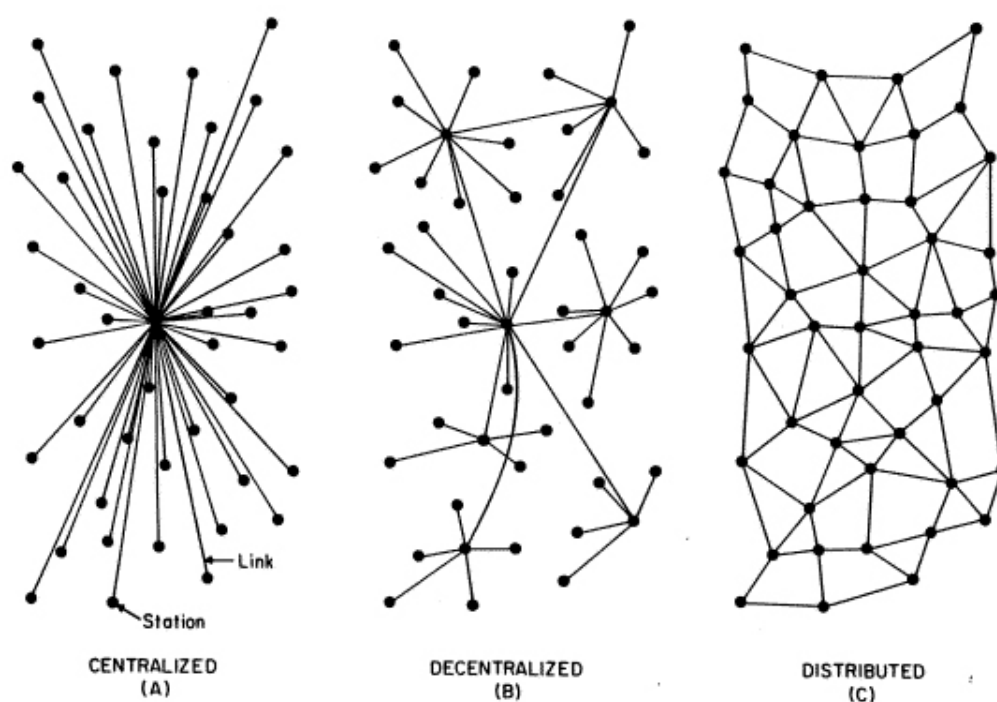


FIG. 1 – Centralized, Decentralized and Distributed Networks

Figura 1.2: Rappresentazione delle possibili configurazioni di network. Internet si basa su una struttura di network distribuita. Immagine tratta da Baran, P., On distributed communications networks, 1962, P-2626, RAND Corporation, <http://www.rand.org/content/dam/rand/pubs/papers/2005/P2626.pdf>.

in modo che la morte di un neurone o la caduta di un collegamento possa essere riparata attraverso l'uso di altri collegamenti" (Berra & Meo 2001, p. 123). La tecnologia principale che governa il funzionamento di Internet, la commutazione di pacchetto, oggi contenuta nel protocollo TCP/IP, sfrutta esattamente questo tipo di struttura per trasmettere informazioni. Infatti l'informazione di base, spedita da un nodo all'altro, viene scomposta in parti (pacchetti) e ciascun pacchetto viene inviato singolarmente sfruttando i numerosi collegamenti che il network presenta tra il punto di partenza e il punto di arrivo. Una volta a destinazione, i pacchetti vengono ricomposti secondo l'ordine originale e l'informazione viene trasmessa. Sfruttando le caratteristiche intrinseche del network, l'informazione può dunque arrivare velocemente attraverso i canali liberi senza intasarli (dal momento che è stata 'scompattata' in piccole porzioni). Il modo attraverso cui le informazioni vengono trasmesse da un nodo all'altro attraverso internet è interessante poiché, come si vedrà in seguito, ha anche degli effetti sulle modalità con cui le

informazioni vengono prodotte e condivise attualmente nel Web.

Due forse le tappe fondamentali (ad oggi) dello sviluppo di Internet: da un lato la nascita e lo sviluppo del WWW (World Wide Web), dall'altro la diffusione dei personal computer.

1.1.3 Il Web

Il Web come lo conosciamo adesso è il cosiddetto Web 2.0, ovvero è nella fase in cui gli applicativi e le tecnologie disponibili (come ad esempio blog, wiki, social network) permettono un elevato livello di interazione con gli utenti. Questo è dovuto principalmente a opportune tecniche di programmazione e alle relative applicazioni afferenti al paradigma del Web dinamico, in contrapposizione al cosiddetto Web statico. Blog, social network, wiki etc. hanno cambiato radicalmente non solo il nostro modo di comunicare, ma hanno anche contribuito a rivedere i processi di produzione dell'informazione.

La situazione attuale non è che l'evoluzione naturale di una delle tappe fondamentali dello sviluppo di internet, che è sicuramente lo sviluppo del meccanismo del World Wide Web: ovvero quel dispositivo che permette, attraverso un link ipertestuale, di passare da una pagina proveniente da un sito in Italia a una pagina proveniente da un qualsiasi altro punto del globo. Questo semplice meccanismo è stato sviluppato nel 1990 presso il CERN da due fisici, Tim Bernes-Lee e Robert Cailliau, nel tentativo di organizzare meglio lo scambio di informazioni e di dati tra i vari laboratori interni al centro di ricerca europeo. "Il Web è una creazione più sociale che tecnica. L'ho progettato per un effetto sociale - aiutare la gente a lavorare insieme - e non come un giocattolo tecnico. Lo scopo finale del Web è quello di sostenere e migliorare la nostra vita in Rete e nel mondo." (Berners-Lee 1999, p.123). La nascita del WWW e del protocollo http permettono, con una certa facilità, idealmente a chiunque di navigare, editare e usufruire di un insieme vastissimo di contenuti collegati tra loro attraverso link. Ciò viene fatto dagli utenti grazie ai programmi detti browser (o navigatori) che ci permettono quotidianamente di navigare nel Web. Tuttavia lo sviluppo del Web è stato possibile sia grazie al contributo degli hacker più esperti e dei programmatori più abili, sia anche grazie alla diffusione tra tutte le persone comuni. Tale diffusione è stata dovuta principalmente alla circolazione dei personal computer che ha permesso di moltiplicare esponenzialmente il numero di nodi connessi alla rete e, di conseguenza, il numero di utenti e di servizi ad essi rivolti.

1.1.4 Il personal computer

Il primo personal computer (il nome fa riferimento all'uso personale che può farne il singolo individuo, in contrasto con i grandi calcolatori presenti nei centri di ricerca o nelle aziende, o con i micro calcolatori presenti sul mercato con funzioni commerciali) fu costruito nel 1976 da Steve Wozniak e venne chiamato Apple I. L'intento di Wozniak era quello di realizzare un computer che potesse essere usato da chiunque e per qualunque scopo, senza che dovesse essere specificata una funzione particolare. Diversamente da quanto si potrebbe credere, principalmente per le fortune successive della Apple (azienda fondata in seguito dal più famoso Steve Jobs), Apple I, stando alle parole dello stesso Wozniak intervistato sull'argomento, "was a total accident of history. I was building projects like terminals for Hewlett-Packard and heard about a club starting up. I designed the Apple I in a closed room in my lab at Hewlett-Packard and in my apartment in Cupertino. I just built it for myself and started showing it off at the Club [the Homebrew Computer Club] by passing out schematics. It was like a science project I was showing off to friends. It wasn't done to be a product to be sold."² (Kennedy 1994). L'aspetto interessante per quanto riguarda la nascita del personal computer, che di lì a poco ebbe una larghissima diffusione, è che il progetto non nasceva con scopi commerciali, ma con scopi puramente intellettuali e scientifici (anche se Wozniak non lavorava all'università e nemmeno era laureato al tempo) nati dalla frequentazione dell'Homebrew Computer Club, un gruppo di hacker che aveva iniziato a incontrarsi con regolarità nella Bay Area a metà degli anni sessanta. Dice ancora Wozniak a questo proposito: "I came from a group that was what you might call beatniks or hippies—a lot of technicians who talked radical about a revolution in information and how we were going to totally change the world and put computers in homes."³ È sorprendente notare che la costruzione del personal computer, che ha contribuito largamente al cambio di passo avvenuto nella produzione di informazione negli ultimi trenta

²«è stato un incidente della storia. Mi stavo occupando di progetti riguardanti dei terminali per Hewlett-Packard e sentì dell'apertura di una nuova associazione. Progettai Apple I chiuso in una stanza nel mio laboratorio a Hewlett-Packard e nel mio appartamento a Cupertino. L'avevo costruito solo per me stesso e iniziai a mostrarlo al Club [il Homebrew Computer Club] distribuendo dei diagrammi illustrativi. Era un progetto scientifico che mostravo agli amici. Non era stato pensato per essere un prodotto da vendere.» [traduzione italiana a cura dell'autore].

³«Venivo da un gruppo di tecnici chiamati beatniks o hippy che facevano discorsi estremisti su una possibile rivoluzione dell'informazione e su come avremmo cambiato radicalmente il mondo e installato computer nelle case.» [traduzione italiana a cura dell'autore].

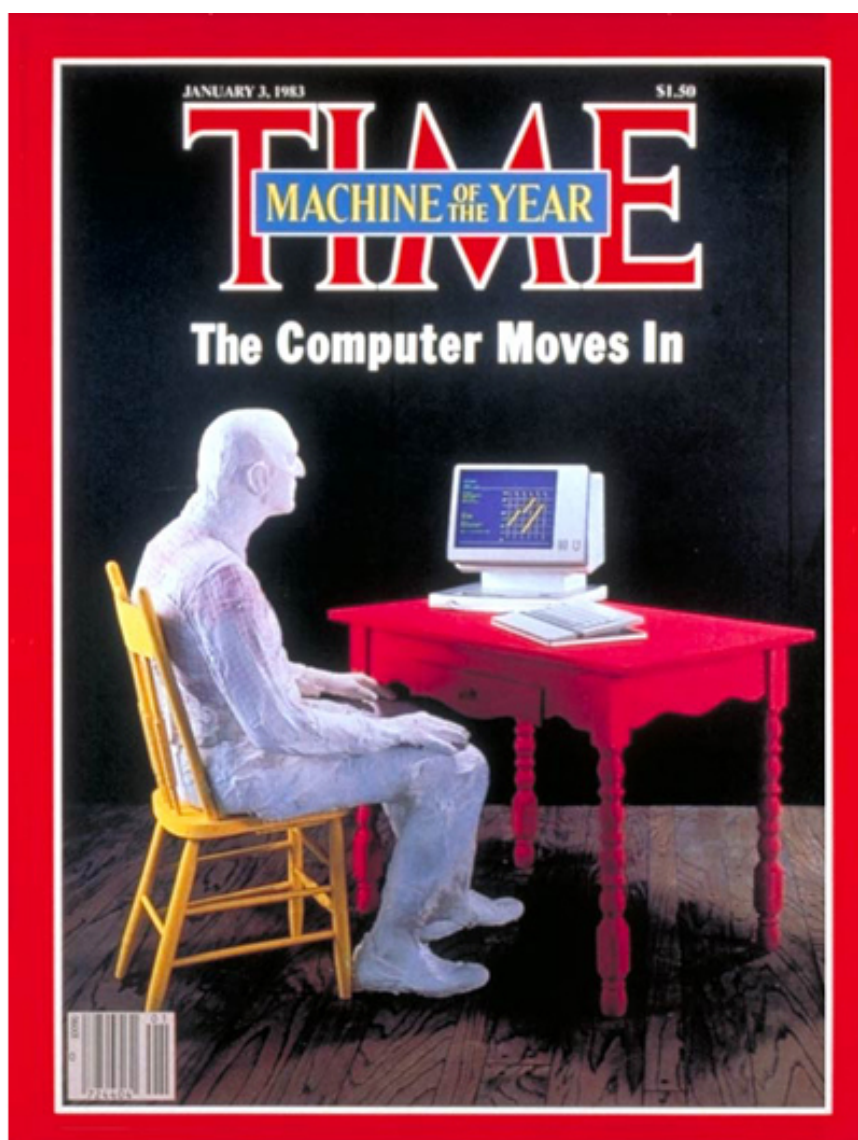


Figura 1.3: La copertina del gennaio 1983 del Time . Come in ogni primo numero dell'anno, il Time elegge la persona più importante dell'anno precedente. In questo caso non si tratta di una persona, ma di una macchina: nel 1982 la macchina dell'anno è il personal computer. Fonte: <http://content.time.com/time/covers/0,16641,19830103,00.html>.

anni (Figura 1.3), non è stata inizialmente un'operazione commerciale⁴ ma

⁴Tutt'altro, se si pensa che Ken Olsen, cofondatore e presidente della della Digital Equipment Corporation, ebbe a dire nel 1977: "Non c'è ragione per cui qualcuno debba volere un computer a casa propria" e che questa affermazione dà l'idea di come l'avvento del personal computer fosse stato interpretato dagli addetti ai lavori.

piuttosto una fortunata e intelligente operazione amatoriale mossa da un più ampio spirito contro culturale che animava la Bay Area in quegli anni.

1.1.5 Hackers

Prima del web, ovvero prima che ci fosse la possibilità per gli utenti comuni di interagire con le tecnologie a disposizione per la creazione e la condivisione di contenuti propri, le persone che si confrontavano con la nascente internet non erano dei semplici utenti ma venivano comunemente indicati come hackers, termine che non ha una accezione negativa (come comunemente si crede) se non per l'uso scorretto che ne è stato fatto nel tempo da parte dei mass media (il termine corretto per indicare il pirata informatico è infatti cracker, non hacker). Infatti in quel particolare periodo molta importanza la ricoprirono non solo le innovazioni tecnologiche, ma anche le persone che vi lavoravano e il loro atteggiamento morale ed etico. Secondo Steven Levy, la storia degli hacker parte dalle prime comunità di studenti appassionati di tecnologia nelle università americane nel secondo dopoguerra (in particolare la Tech Model Railroad Club TMRC, un gruppo di appassionati di modellismo ferroviario del MIT) la cui curiosità e applicazione nei confronti delle tecnologie elettroniche si diffuse in tutto il mondo (basti pensare ai radioamatori o agli appassionati di elettronica) (Levy 1996b). Oltre che prettamente orientata alla tecnica e alla tecnologia, l'espressione dei primi hacker fu sicuramente anche di natura "culturale e politica, e si riferisce ai valori progettati dai movimenti sociali alla fine degli anni Sessanta e dei primi Settanta in Europa e America. Questi movimenti furono fondamentalmente libertari [...]. Questi movimenti furono fondamentalmente culturali perché non si concentrarono sulla presa del potere statale (a differenza della maggior parte dei loro predecessori nel secolo) o sulla redistribuzione della ricchezza. Invece essi hanno agito sulle categorie dell'esperienza respingendo le istituzioni ufficiali, alla ricerca di nuovi significati di vita e, conseguentemente, di una ridefinizione dei contratti sociali tra individuo e stato, e tra individuo e mondo aziendale." (Castells 2001).

Il contatto tra hacker (appassionati di alta tecnologia) e movimenti libertari (il cui appartenente medio potrebbe essere identificato in un hippy delle prime ore) può anche sembrare forzato, o non del tutto esplicito. Scrive a tal proposito Kevin Kelly, fondatore della rivista Wired: "Le origini della wired generation (la generazione connessa) e della cultura dei computer dei capelloni (pensiamo all'UNIX open source) le trovate proprio nei fuggiaschi della controcultura degli anni Settanta (Figura 1.4). Come ricorda Stewart Brand, il fondatore (anche lui hippy) del Whole Earth Catalog, «Fai la tua cosa' si traduceva facilmente in 'Avvia la tua impresa'». Ho perso il con-



Figura 1.4: I movimenti libertari portarono con sé nuovi modi di abitare e di vivere. In alcuni casi questi stili di vita alternativi erano accompagnati da una forte componente tecnologica che si esprimeva nella autocostruzione delle nuove abitazioni: “Proposte all’industria americana, che di fatto le guarda con scetticismo, le cupole geodetiche fanno fatica a decollare. Vengono, però, accolte con entusiasmo dai giovani [...] Nel 1966 a Città di Trinidad, nel Colorado, dieci ragazzi e tre ragazze, le utilizzano per realizzare una comune dal nome emblematico di Drop City. I materiali usati sono semplici e economici [...] Drop City diventa ben presto un modello per una generazione di ragazzi che al Vietnam e al carrierismo mostra di preferire l’amicizia, il libero amore, la musica.” (Prestinenzza Puglisi 2013).

to delle centinaia di persone nella cerchia delle mie conoscenze che hanno lasciato le comuni per avviare, alla fine, società high tech nella Silicon Valley. Ormai è quasi un cliché: «da scalzo a miliardario», proprio come Steve Jobs.” (Kelly 2011). Non fu un caso dunque che proprio l’autore del Catalog, Stewart Brand (con Steven Levy) fosse tra gli organizzatori della prima Hacker Conference nel 1984 (Brand fu anche co-fondatore di The WELL, The Whole Earth ’Lectronic Link, una delle comunità virtuali più antiche e numerose). Proprio alla prima Hacker Conference del 1984 “Burrell Smith, l’hacker che stava dietro il computer Apple della Macintosh, definì il termine nel modo seguente: «Gli hacker possono fare qualsiasi cosa e restare sempre hacker. Per essere un hacker, l’alta tecnologia non è assolutamente necessaria. Penso piuttosto che l’essere hacker abbia a che fare con l’abilità e la dedizione per ciò che si fa». Nella guida *Come diventare hacker*, Raymond osserva che “ci sono persone che applicano l’attitudine hacker a cose diverse dal software, come l’elettronica e la musica, in realtà la si può trovare ai più alti livelli di qualsiasi scienza o arte” (Himanen 2001, p. 17). Proprio perché

a forte connotazione culturale e politica l'attitudine hacker è, secondo la loro stessa comunità, portatrice di un'etica propria, l'etica hacker. Secondo il *Jargon File*⁵, un documento online che comprende e spiega il lessico hacker, l'etica hacker si basa sulla "credenza che la condivisione di informazioni è un bene fortemente positivo ed è un dovere etico degli hacker condividere la loro esperienza scrivendo codice open-source, facilitando l'accesso alle informazioni e a risorse computazionali quando possibile". L'etica hacker non è esattamente oggetto di questa tesi, e maggiori informazioni possono essere trovate nel libro di Pekka Himanen *L'etica Hacker* del 2001 (che è anche la fonte principale di questo paragrafo). Ciò che però è sicuramente interessante notare è che il fenomeno degli hacker e dell'hacking è un fenomeno nuovo, difficilmente inquadrabile in un'etica del lavoro di stampo capitalista come la nostra. Per meglio comprenderla si può risalire ai precedenti storici dell'etica hacker che sono l'accademia e l'etica scientifica: "quando il famoso sociologo della scienza Robert Merton descrisse lo sviluppo dell'etica scientifica nel Rinascimento, affermò che una delle sue fondamenta era il 'comunismo', ovvero l'idea che la conoscenza scientifica dovesse essere pubblica, un'idea che il Rinascimento ha fatto rivivere dall'etica accademica della prima comunità scientifica, l'Accademia di Platone, che si basava sull'idea di *synusia*, l'azione concertata nella quale la conoscenza veniva condivisa liberamente." (Himanen 2001, p. 45).

1.2 Dal free software all'Open Source

1.2.1 Free software

Non si può parlare di free software senza accennare all'inventore del termine, il fisico (ma più conosciuto per le sue doti di hacker e informatico) Richard Stallman. È infatti a lui che deve essere attribuito il merito di aver coniato il termine free software e di aver sviluppato gli strumenti che ne permettono lo sviluppo e l'utilizzo, come ad esempio la licenza GNU-GPL. Impiegato presso il centro di ricerca sull'Intelligenza artificiale al MIT dal 1971 al 1984, Stallman (che si autodefinisce l'ultimo hacker) si scontrò presto con le prime licenze software protette da copyright che non permettevano all'utente finale di apportare modifiche adattando il software alle proprie necessità. Abituato

⁵Il *Jargon File* è un documento originariamente redatto da Raphael Finkel della Stanford University e attualmente mantenuto da Eric S. Raymond, un esponente della cultura hacker nel mondo. Esso è essenzialmente un vocabolario del gergo usato dagli hacker e dai professionisti dell'IT, ma contiene anche definizioni e regole di buona educazione da rispettare in rete (netiquette). L'attuale versione si trova all'indirizzo <http://www.catb.org/jargon/>.

a far parte di una comunità “in cui ci si scambiavano i programmi, che esisteva già da molti anni. La condivisione del software non si limitava alla nostra comunità; è una cosa vecchia quanto i computer, proprio come condividere le ricette è antico come l'arte culinaria. Ma noi lo facevamo più di chiunque altro” (Stallman 2003, p. 11).

Stallman abbandonò il laboratorio del MIT non appena si rese conto che il software proprietario stava ormai permeando il suo lavoro e che non era più possibile utilizzare liberamente le macchine del laboratorio. Decise così di intraprendere la compilazione di un nuovo sistema operativo, denominato GNU, che potesse essere totalmente libero. Attuò ciò attraverso la creazione di una apposita fondazione no-profit, la Free Software Foundation (FSF), la quale aveva il compito di occuparsi dello sviluppo del software libero. Software libero, secondo l'interpretazione di Stallman, ha una particolare accezione: “Il ‘software libero’ è una questione di libertà, non di prezzo. Per capire il concetto, bisognerebbe pensare alla ‘libertà di parola’ e non alla ‘birra gratis’ [il termine ‘free’ in inglese significa sia ‘gratuito’ che ‘libero’, in italiano il problema non esiste]. L'espressione ‘software libero’ si riferisce alla libertà dell'utente di eseguire, copiare, distribuire, studiare, cambiare e migliorare il software. Più precisamente, esso si riferisce a quattro tipi di libertà per gli utenti del software:

Libertà di eseguire il programma, per qualsiasi scopo (libertà 0).

Libertà di studiare come funziona il programma e adattarlo alle proprie necessità (libertà 1). L'accesso al codice sorgente ne è un prerequisito.

Libertà di ridistribuire copie in modo da aiutare il prossimo (libertà 2).

Libertà di migliorare il programma e distribuirne pubblicamente i miglioramenti, in modo tale che tutta la comunità ne tragga beneficio (libertà 3). L'accesso al codice sorgente ne è un prerequisito” (Stallman 2003, p. 59).

Se si pensa che la cultura hacker, cui Stallman fa riferimento, derivava direttamente dai movimenti libertari degli anni Sessanta all'interno dei quali la componente etica e morale non è affatto da sottovalutare, non è un caso che la definizione di software libero affronti direttamente il concetto di libertà, declinandone gli attributi rispetto al software e ai suoi usi. L'invenzione più interessante di Stallman fu però quella del Copyleft (anche detto permesso d'autore), ovvero del regime giuridico all'interno del quale è possibile mantenere tutte le caratteristiche del software libero e quest'ultimo sia considerato tale. Ciò avvenne principalmente attraverso lo sviluppo di una apposita licenza (la licenza GNU-GPL) atta a garantire lo status libero al software che veniva distribuito. Nel particolare “il permesso d'autore [copyleft] usa le leggi sul diritto d'autore [copyright], ma le capovolge per ottenere lo scopo opposto: invece che un metodo per privatizzare il software, diventa infatti un mezzo per mantenerlo libero. Questo tipo di licenza è l'innovazione giuridica

più importante del movimento free software. [...] La GPL richiede a chiunque modifichi il software e ne distribuisca una versione modificata di rilasciarla sotto lo stesso tipo di licenza che ha il software originale.” (Benkler 2007, p. 81).

“Il succo dell’idea del copyleft consiste quindi nel dare a chiunque il permesso di eseguire il programma, copiare il programma, modificare il programma e distribuirne versioni modificate, ma senza dare il permesso di aggiungere restrizioni. In tal modo, le libertà essenziali che definiscono il “free software” sono garantite a chiunque ne abbia una copia, e diventano diritti inalienabili. Perché un permesso d’autore sia efficace, anche le versioni modificate devono essere libere.” (Stallman 2003, p. 22). Il concetto di copyleft, oltre a garantire la maggior parte delle licenze di software libero e open source, è alla base delle attuali licenze Creative Commons che governano la distribuzione dei contenuti di Wikipedia e di tanti altri servizi online.

Un evento piuttosto importante caratterizzò lo sviluppo e la diffusione del software libero, e fu l’avvento di Linux nel 1991. Uno studente finlandese, Linus Torvalds, cominciò a sviluppare un suo personale sistema operativo, denominato Linux, e ne condivise attraverso Internet il codice sorgente invitando tutti gli appassionati a provarlo sui propri calcolatori e a fornire commenti e feedback sul suo funzionamento. Ma quello che ottenne fu molto di più: moltissimi sviluppatori risposero al suo invito e cominciarono a collaborare con lui per migliorare ed espandere il codice sorgente. Così nell’arco di pochi anni, grazie al telelavoro collettivo di tutti gli appassionati guidati dallo stesso Torvalds, venne sviluppato un sistema competitivo per affidabilità e sicurezza secondo modalità di lavoro del tutto inconsuete che di lì a poco avrebbero cambiato il mondo dell’informatica e dei contenuti digitali. Lo sviluppo di Linux si basava (e si basa ancora oggi) sul principio che il codice sorgente (source) potesse essere condiviso liberamente tra gli interessati e, poiché era visibile a tutti e sostanzialmente aperto (open), le persone coinvolte potevano effettuare modifiche per rendere il codice più robusto ed efficiente, aggiungere nuove funzionalità, riparare eventuali errori e risolvere bug. In pochi anni e a costi irrisori (il costo di trasmissione di dati a carico degli utenti stessi, comunque trascurabile) era stato creato un prodotto che non solo era affidabile e robusto, ma anche concorrenziale rispetto ai sistemi operativi commerciali (sviluppati grazie a ingenti investimenti delle software house) e soprattutto in continuo sviluppo, miglioramento ed espansione. Per confermare il successo di Linux basta pensare al fatto che la maggior parte di server e supercomputer utilizzano Linux come sistema operativo al posto di altro software commerciale.

1.2.2 Open Source

*"The hacker movement referred to by the term open-source has burst into public consciousness in the last few years due to its spectacular success in the production of reliable and robust software. Perhaps the most obvious symptom of this success is the fact that open-source software in several key areas (operating systems, server software) is the only serious alternative to the domination of the market by large corporations like Microsoft. Its paradigm of software production, collectively-created programs in a process where users are also (to different degrees) developers, has gone beyond the expectations of most analysts, and taken by surprise most corporate managers many of which (at corporations like IBM and SUN) are rapidly switching from proprietary standards to open systems."*⁶

(DeLanda 2001)

Nel precedente paragrafo si è parlato principalmente di free software, mentre in questo paragrafo si affronterà il tema dell'Open Source. In realtà i due termini, seppure differenti, vengono spesso utilizzati con una certa ambiguità per indicare la stessa cosa. Secondo Stallman "Software libero (free software) e sorgente aperto (Open Source) descrivono più o meno la stessa categoria di software, ma dicono cose differenti sul software e sui valori. Il progetto GNU continua a usare il termine 'software libero' per esprimere l'idea che la libertà sia importante, non solo la tecnologia." (Stallman 2003, p. 40). Se dunque free software presuppone anche delle caratteristiche etiche riguardo la produzione e la distribuzione del software, Open Source ne indica unicamente il metodo di sviluppo. L'uno non esclude automaticamente l'altro, ma non necessariamente l'uno corrisponde anche all'altro. A partire dal 1998 la comunità ha cominciato a dividersi sull'uso dei due termini. Se da un lato il termine free software resiste tra i puristi del software libero, il termine Open Source ha avuto più fortuna sia dal punto di vista commerciale che dal punto di vista mediatico, arrivando a essere un appellativo di successo per il software (si pensi ad esempio al sistema operativo mobile Android). "Il termine

⁶"Il movimento hacker a cui si riferisce il termine "open source" ha fatto irruzione nella coscienza pubblica negli ultimi anni grazie al suo grande successo nella produzione di software affidabili e robusti. Forse il sintomo più evidente di questo successo risiede nel fatto che il software open-source è l'unica alternativa seria in molti settori chiave (sistemi operativi, server software) al predominio nel mercato di grandi aziende come Microsoft. Il suo paradigma di produzione di software, di programmi creati collettivamente in un processo in cui gli utenti sono anche (in diversa misura) gli sviluppatori, è andato aldilà delle aspettative di molti analisti, e ha colto di sorpresa la maggior parte dei manager aziendali, molti dei quali infatti (in multinazionali come IBM e SUN) sono passati velocemente da standard proprietari a sistemi aperti." [traduzione italiana a cura dell'autore].

open source software venne introdotto al posto di free software per neutralizzarne i connotati politici. Era semplicemente una forma di organizzazione della produzione di software che poteva essere più efficace di quella basata sul mercato. [...] La vasta adozione dell'open source nei circuiti burocratici ed economici ha permesso al software libero di lasciare i margini del mondo informatico, conquistando il centro del dibattito sulle alternative allo stato di cose presente" (Benkler 2007, p. 84). Inoltre bisogna considerare il fatto che il termine Open Source ha avuto fortuna anche in altri campi applicativi, non solo nel campo del software (come si vedrà nel prossimo capitolo). Come detto in precedenza, Linux è forse il primo software open source di successo e ciò significa che non solo con Linux si è verificata la bontà di un metodo di sviluppo software non tradizionale (in realtà, seguendo la storia degli hacker, il software è sempre stato oggetto di mutuo scambio) e commerciale, ma si è anche riusciti a dimostrare come il paradigma Open Source sia valido tanto quanto i metodi di sviluppo chiusi adottati dalle software house. Ciò ha contribuito ad affermare che la passione e gli interessi personali, se opportunamente gestiti e veicolati, hanno la possibilità di tenere testa a investimenti anche cospicui delle grandi software house. È infatti piuttosto chiaro che l'Open Source funziona attraverso il contributo di un numero variabile di persone interessate al progetto e che, tramite un mutuo scambio di codice, pareri e suggerimenti il software viene sviluppato collettivamente e costantemente migliorato e aggiornato. Ciò che è meno chiaro è come effettivamente il progetto di software open source debba essere gestito in modo che gli sforzi dei singoli utenti siano ottimizzati e costantemente rivolti allo sviluppo, evitando di ritrovarsi in una babele di programmatori in sostanziale concorrenza tra loro.

Se dal punto di vista teorico il funzionamento è piuttosto chiaro, dal punto di vista pratico alcune interessanti indicazioni ci vengono fornite da Eric S. Raymond nel suo testo *La cattedrale e il bazaar*. Raymond, particolarmente colpito dal successo di Linux, decise di cominciare a gestire lo sviluppo Open Source di un client mail (Fetchmail, un analogo dei contemporanei Thunderbird e Outlook) e di sperimentare alcune teorie circa lo sviluppo del software in modo aperto. Scrive infatti nell'introduzione del suo testo, riferendosi al titolo dello stesso: "Linux stravolse gran parte di quel che credevo di sapere. Per anni avevo predicato il vangelo Unix degli strumenti agili, dei prototipi immediati e della programmazione evolutiva. Ma ero anche convinto che esistesse un punto critico di complessità al di sopra del quale si rendesse necessario un approccio centralizzato e a priori. Credevo che il software più importante (sistemi operativi e strumenti davvero ingombranti come Emacs) andasse realizzato come le cattedrali, attentamente lavorato a mano da singoli geni o piccole bande di maghi che lavoravano in splen-

dido isolamento, senza che alcuna versione beta vedesse la luce prima del momento giusto. Rimasi non poco sorpreso dallo stile di sviluppo proprio di Linus Torvalds – diffondere le release presto e spesso, delegare ad altri tutto il possibile, essere aperti fino alla promiscuità. Nessuna cattedrale da costruire in silenzio e reverenza. Piuttosto, la comunità Linux assomigliava a un grande e confusionario bazaar, pullulante di progetti e approcci tra loro diversi (efficacemente simbolizzati dai siti contenenti l'archivio di Linux dove apparivano materiali prodotti da chiunque). Un bazaar dal quale soltanto una serie di miracoli avrebbe potuto far emergere un sistema stabile e coerente.” (Raymond 1998, p. 1) Il tipo di approccio a bazar è per l'appunto l'oggetto delle sue ricerche, dalla cui sperimentazione Raymond enuncia 19 regole d'oro, che vengono messe in pratica e sperimentate attraverso lo sviluppo del sopracitato Fetchmail. Di queste 19 regole, di seguito verranno trascritte e brevemente commentate le principali e più significative, indicate con il numero presente nel saggio.

“1) *Ogni buon lavoro software inizia dalla frenesia personale di uno sviluppatore.*” Lo sviluppo di nuovo software innovativo non può essere unicamente la voglia di risultare competitivi nel mercato, ma molto spesso è collegato alla necessità di risolvere problemi contingenti anche personali. In altre parole, la necessità è la madre di tutte le invenzioni (Berra & Meo 2001).

“2) *I bravi programmatori sanno cosa scrivere. I migliori sanno cosa riscrivere (e riusare).*” Ovvero Nulla si crea, nulla si distrugge, tutto si trasforma. Nulla si crea ex-novo nel mondo della tecnologia moderna (caratterizza da un'enorme complessità intrinseca) ma molto spesso si parte da un modello preesistente con l'obiettivo di migliorarlo adattandolo alle proprie necessità.

“6) *Trattare gli utenti come co-sviluppatori è la strada migliore per ottenere rapidi miglioramenti del codice e debugging efficace.*”

“7) *Distribuisci presto. Distribuisci spesso. E presta ascolto agli utenti.*”

“8) *Stabilita una base di beta-tester e co-sviluppatori sufficientemente ampia, ogni problema verrà rapidamente definito e qualcuno troverà la soluzione adeguata.*”

“9) *La cosa migliore, dopo l'aver buone idee, è riconoscere quelle che arrivano dagli utenti. Qualche volta sono le migliori.*”

Queste quattro regole costituiscono il vero nodo innovativo portato da Linus Torvalds nello sviluppo di Linux. Tale modello di collaborazione basato sull'integrazione dei ruoli degli utilizzatori e degli sviluppatori è il primo e necessario passo per attivare un sistema aperto (o un sistema bazaar) in cui la partecipazione di più attori sia veramente un valore aggiunto fondamentale.

“19) *Stabilito che il coordinatore dello sviluppo abbia a disposizione un medium almeno altrettanto affidabile di Internet, e che sappia come svolgere*

il ruolo di leader senza costrizione, molte teste funzionano inevitabilmente meglio di una sola.”

In questo caso il commento più efficace è quello dato dallo stesso Raymond: “Penso che il futuro del software open source apparterrà sempre più alle persone che sanno come giocare al gioco di Linus, persone che si lasceranno alle spalle la cattedrale per entrare nel bazaar. Ciò non significa che non saranno più importanti le visioni e l’intelligenza individuali; credo piuttosto che la punta avanzata del software open source apparterrà a quanti sapranno muovere da visioni e intelligenza individuali per poi amplificarle tramite l’effettiva costruzione di comunità volontarie d’interesse. E forse ciò vale non soltanto per il futuro del software open source. Nessuno sviluppatore in ‘closed -source’ potrà mai pareggiare la fucina di talenti che la comunità Linux è in grado di riunire per affrontare un problema. Sono davvero in pochi a potersi permettere di assumere le oltre duecento persone che hanno contribuito a Fetchmail! Forse alla fine la cultura dell’open source trionferà non perché la cooperazione sia moralmente giusta o perché il software ‘costretto’ sia moralmente sbagliato (dando per scontato che si creda a quest’ultima affermazione, cosa che né io né Linus facciamo), ma semplicemente perché il mondo ‘closed -source’ non è in grado di vincere la corsa agli armamenti dell’evoluzione contro quelle comunità open source capaci di affrontare un problema con tempi e capacità superiori di diversi ordini di grandezza.” (Raymond 1998, p. 13)

1.2.3 Anarchia organizzata

“Without global control from the top, the Linux project has achieved a fine balance between evolution and self-organization, order and disorder. Stuart Kauffman argues that complex systems come to the edge of chaos through the same processes that guide Darwinian evolution, namely, natural selection. Systems must evolve for a right amount of evolvability; systems that do not find the sweet spot simply die out. But once the system is at the edge of chaos, we are bound to see surprises. Linux, like the eye, is one such system that has come to dazzle us all.”⁷
(Kuwabara 2000)

⁷“Senza un controllo globale dall’alto, il progetto Linux ha raggiunto un buon equilibrio tra evoluzione e auto-organizzazione, tra ordine e disordine. Stuart Kauffman sostiene che sistemi complessi arrivano sull’orlo del caos attraverso gli stessi processi che guidano l’evoluzione darwiniana, ovvero attraverso la selezione naturale. I sistemi devono evolversi fino a un certo livello di sviluppo; i sistemi che non trovano il punto giusto semplicemente muoiono. Ma è quando il sistema è sull’orlo del caos che abbiamo delle sorprese. Linux, come l’occhio, è uno di quei sistemi che è arrivato a stupire tutti noi.” [traduzione italiana a cura dell’autore].

“Lo spirito Open Source postula una sorta di cyberanarchia che è deregolata, come si conviene a tutte le forme di anarchia, ma anche ispirata a ideali di collaborazione, solidarietà e disinteresse. Contro ogni attesa, e malgrado enormi gli enormi interessi economici in gioco, è in questo spirito libertario che dobbiamo alcune delle più brillanti invenzioni nel campo delle nuove tecnologie, dal protocollo http al sistema operativo Linux.”

(Carpo 2009b)

Molti autori si sono concentrati sul funzionamento di Linux e sul suo tipo di organizzazione, la quale, al pari delle tipologie organizzative tradizionali presenti nell'economia aziendale, è in grado di produrre risultati concorrenziali, sebbene non agisca all'interno delle stesse dinamiche di mercato dei suoi concorrenti. Difatti “secondo le nostre normali assunzioni sui progetti volontari e sui processi a produzione decentrata che non hanno una gestione manageriale, questo modello non dovrebbe avere successo. Ma l'ha avuto e continua ad averlo.” (Benkler 2007, p. 83). Questo sistema di produzione sociale orizzontale è stato studiato e affrontato da diversi autori di cui si proverà ad esporre brevemente le tesi. Tuttavia si ritiene che tutte le ipotesi presentate siano valide poiché sono proposte da persone con formazione e interessi diversi, le quali leggono l'Open Source dal punto di vista della loro disciplina e danno letture differenti ma non per questo incompatibili. Meo parla di una sorta di anarchia organizzata o di giungla, rifacendosi a studi di un gruppo di ricercatori statunitensi e norvegesi dell'università di Bergen, sostenendo la funzionalità del disordine. “Per certi versi infatti il sistema organizzativo di Linux presenta delle analogie con le teorizzazioni dell'organizzazione come giungla o come anarchia organizzata, nel senso che spesso sono i problemi, contrariamente alla logica tradizionale del problem solving, che vanno in cerca delle soluzioni” (Berra & Meo 2001, p. 117). Kuwabara si concentra invece sulla complessità intrinseca del sistema Linux e su come l'accettazione e non la riduzione della complessità abbia portato ad avere buoni risultati: “In short, it is psychologically appealing to view aggregate phenomena in terms of simple universal laws of nature, but such a point of view neglects the true complexity of reality. The case of Linux suggests, instead, that aggregate behaviors are better understood as emergent properties - properties that emerge out of micro-level interactions among constituents - than as qualities intrinsic to the system and independent of its constituent parts. Moreover, systems that allow open and active interaction can be more adaptive and more dynamic than closed systems that restrict their constituents. In the case of Linux, nothing in its architectural design was particularly innovative or revolutionary. [...] What set Linux apart from other operating systems in the level of quality and performance, Raymond

submits, is not anything inherent in the original architecture, but the open and evolutionary development.”⁸(Kuwabara 2000).

Benkler, parlando di organizzazione orizzontale, si sofferma sulle tre caratteristiche che la contraddistinguono. In primis la presenza e la diffusione dei personal computer e altri device (“primo, il macchinario fisico per partecipare alla produzione di informazione e cultura è distribuito in modo praticamente universale tra la popolazione delle economie avanzate”) che attraverso Internet sono collegati in rete e permettono non solo di produrre, ma anche di condividere ciò che viene prodotto a costi trascurabili. In seguito c’è la questione legata alla materia prima che, differentemente dalla produzione tradizionale degli ultimi secoli, non si basa sul capitale privato: “Secondo, le materie prime dell’economia dell’informazione, diversamente da quelle dell’economia industriale, sono beni pubblici: l’informazione, la conoscenza e la cultura esistenti”. E infine la possibilità di partecipazione da parte degli utenti favorita dalla capacità dell’organizzazione di assorbire contributi di grana fine, che richiedono minori sforzi in termini di tempo e risorse: “Terzo, architetture tecnologiche, modelli organizzative dinamiche sociali di scambio e produzione di informazione su internet si sono sviluppati in modo da permettere di strutturare la soluzione dei problemi in forma altamente modulare.”(Benkler 2007, p. 134).

In ogni caso il sistema organizzativo che sta alla base dell’Open Source può essere considerato alla stregua di una learning organization, ovvero una struttura che facilita l’apprendimento dei suoi membri ed è in un perenne stato di trasformazione o evoluzione (Senge 1992).

1.2.4 Reputazione e collaborazione

“Le competenze individuali diventano quindi il nucleo economico della produzione di cultura e informazione, che ha la priorità sulla capa-

⁸“In breve, è psicologicamente interessante per vedere i fenomeni di aggregazione in termini di semplici leggi universali della natura, ma un tale punto di vista omette la vera complessità della realtà. Il caso di Linux suggerisce invece che i comportamenti di aggregazione siano meglio comprensibili se considerate come proprietà emergenti – proprietà che emergono da interazioni di micro livello tra i costituenti – piuttosto che come qualità intrinseche al sistema e indipendenti dalle sue parti costituenti. In aggiunta a ciò, i sistemi che permettono un’interazione aperta e attiva possono adattarsi meglio ed essere più dinamici rispetto ai sistemi chiusi, che invece limitano le loro parti costituenti. Nel caso di Linux, nulla nel suo progetto architettonico è stato particolarmente innovativo o rivoluzionario. [...] Raymond sostiene che cosa ha differenziato Linux rispetto agli altri sistemi operativi, in termini di qualità e performance, non ha a che vedere con l’architettura originale ma dipende dal suo sviluppo aperto e in continua evoluzione.” [traduzione italiana a cura dell’autore].

ciò di raccogliere capitale finanziario. Parte di questa competenza è scambiata sui mercati come lavoro creativo, e continuerà ad esserlo. Tuttavia, grazie alla sua liberazione dai ceppi del capitale fisico, le persone creative possono prendere parte a una gamma più ampia di pratiche di produzione di cultura e informazione rispetto al tempo in cui, oltre a creatività, esperienza, sapere e tempo, per produrre informazione c'era bisogno di avere milioni di dollari. [...] Nell'economia fisica queste relazioni erano relegate quasi completamente al di fuori del sistema di produzione, mentre l'economia dell'informazione in rete reca con sé la promessa di proiettare la ricchezza della vita sociale al centro dell'economia e della produzione.”

(Benkler 2007, p. 66)

“La proprietà che emerge da un sistema di menti umane in comunicazione fra loro è il fatto che esse realizzano opere per il reciproco godimento e per vincere il loro penoso senso di solitudine.”

(Moglen 1999)

Una delle domande principali che ci si pone nel momento in cui si affronta il tema dell'Open Source è il tentativo di capire quali siano le motivazioni che spingono le persone a parteciparvi. Infatti non si è all'interno della normale produzione industriale, dove il lavoro viene normalmente remunerato, ma ci si trova in un sistema di produzione sociale, dove la remunerazione non ha luogo, o perlomeno non ha luogo come tradizionale transizione di denaro tra datore di lavoro e lavoratore. Molti autori hanno dato la loro risposta e molti sono i contributi scientifici che possono essere presi in esame. Di questi, si farà un breve accenno a quelli più significativi e interessanti per il prosieguo della tesi, consci che questa breve trattazione possa non essere considerata esaustiva.

La prima risposta si rifà alla teoria del dono, ovvero la teoria secondo cui “ciò che dà origine al software libero è un atto di liberalità da parte di chi mette il prodotto in rete (il donatore), a cui corrisponde una libertà di risposta da parte di chi riceve. I tre obblighi del dare, ricevere e restituire impliciti nel dono non sono garantiti formalmente, ma sono obblighi morali che mantengono un contenuto di libertà” (Berra & Meo 2001).

Altra risposta possibile è quella basata sul fattore della reputazione, ovvero che la partecipazione a questo tipo di esperienze in maniera produttiva porti a un riconoscimento personale all'interno della comunità: “the norm of cooperation is shaped by reputation game because people compete for peer recognition.”⁹(Kuwabara 2000).

⁹“La regola della cooperazione si basa sul gioco della reputazione perché le persone competono per il riconoscimento tra pari.” [traduzione italiana a cura dell'autore].

Oltre a queste due risposte ve ne è una terza, forse più semplicistica, ma per certi versi anche più efficace. Secondo Eben Moglen la risposta “è nella natura umana. È come chiedersi perché Figaro canta, perché Mozart compose la musica che lui canta, e perché tutti noi parliamo: perché possiamo. Homo ludens, stringi la mano all’ Homo Faber. La condizione sociale di interconnessione globale che chiamiamo Internet rende possibile a tutti noi di essere creativi in modi nuovi e mai sognati. A meno che non lasciamo che la ‘proprietà’ interferisca.” (Moglen 1999). Ed è questa forse la risposta più esaustiva. Non si spiegherebbe sennò il lavoro enorme svolto da tutti i contributori di Wikipedia (la riposta del dono è accettabile, ma prevede la proprietà di ciò che si dona e soprattutto che questa passi a qualcun altro, quando nella realtà la proprietà dell’informazione non è uno dei fattori in gioco essendo essa libera); né tantomeno quello svolto dai redattori della documentazione che accompagna il software (che fondamentalmente non viene considerata una attività di rilievo, anche se fatta a regola d’arte, non comporta infatti particolare considerazione da parte dei programmatori e quindi non porta a ottenere alcun tipo di reputazione). La voglia di creare e comunicare, che si combina con l’esperienza sociale e culturale che si esplicita nel Web, ha come effetto quello di spingere ognuno di noi a parlare di ciò che pensiamo possa interessare anche altri. E questo va esattamente contro il sistema di produzione di massa dell’informazione poiché al posto di poche voci (o emettitori di informazioni) si creano e si creeranno tutta una serie di mercati di nicchia all’interno dei quali vigeranno dei sistemi di produzione non commerciali. Sempre più informazione verrà prodotta da sempre più attori a costi sempre più marginali. Ci troveremo dunque in una situazione nella quale “i comportamenti sociali che tradizionalmente erano legati ai margini dell’economia capitalista sono diventati elementi fondamentali delle economie avanzate. I comportamenti non commerciali stanno diventando fattori determinanti per la produzione dell’ambiente informazionale e culturale.” (Benkler 2007).

1.2.5 Modularità e ricombinazione

“Nella ricombinazione di tutta l’informazione e di tutta la comunicazione esistente sulla base di scopi specifici decisi in tempo reale da ciascun utente/produttore dell’ipertesto. La ricombinazione è fonte di innovazione, in particolare se i prodotti che genera diventano a loro volta sostegni per una ulteriore interazione, in una spirale di informazioni sempre più significative. La generazione di nuove conoscenze richiederà sempre l’applicazione di una teoria dell’informazione ricombinante. Le possibilità di sperimentare con questa caratteristica che ha l’informazione da una molteplicità di fonti ampliano notevolmente sia il campo

della conoscenza, sia le connessioni che possono essere fatte tra ambiti differenti - proprio come la fonte di innovazione della conoscenza veniva descritta da Kuhn nella teoria delle innovazioni scientifiche.”

(Castells 2001)

È molto interessante osservare come i processi di produzione sociale che si sono sviluppati grazie ad internet, e che Benkler descrive così bene nel suo libro *La ricchezza della rete*, siano profondamente collegati al tipo di infrastruttura che ne permette la diffusione e la crescita. Non a caso infatti la produzione sociale, per essere efficace, si basa su due concetti fondamentali: il primo è la modularità, il secondo è la ricombinazione. Vediamo meglio di cosa si tratta.

Nel paragrafo precedente si è visto come uno dei fattori che garantiscono l'efficacia della produzione sociale sia la possibilità di strutturare la soluzione dei problemi in forma altamente modulare. Ciò significa che la riduzione a entità modulari di qualsivoglia problema ne favorisce una risoluzione di tipo orizzontale poiché ogni attore può scegliere il modulo su cui operare in maniera indipendente, conscio del fatto che il suo tassello andrà a comporre un quadro complessivo risolutivo (frutto di altri tasselli che sono frutto a loro volta del lavoro di altri). Ciò permette alcuni vantaggi: da un lato l'utente è libero di scegliere secondo il suo gusto personale o secondo la sua esperienza o le sue capacità, dall'altro può scegliere secondo la sua disponibilità di tempo e voglia di partecipare. “La ‘modularità’ è la proprietà che descrive il grado con cui un progetto può essere spezzettato in componenti più piccole, o moduli, che possono essere prodotti indipendentemente e poi riasssemblati in modo coerente. Se i moduli sono indipendenti, i singoli collaboratori sono messi nella condizione di scegliere come e quando contribuire, in maniera indipendente l'uno dall'altro. In questo modo vengono massimizzate l'autonomia e la flessibilità nel definire natura, grandezza e durata della partecipazione al progetto.” (Benkler 2007, p.128). Volendo fare un piccolo esempio si pensi all'utente di Wikipedia interessato a contribuire: egli potrà scegliere quale argomento approfondire, su quale specifica voce intervenire, quante voci modificare, creare o implementare, il tutto seguendo le sue inclinazioni culturali, a partire dalla lingua di interesse fino al più piccolo particolare. Potrà anche solo modificare o inserire la data precisa di un evento storico, potrà contribuire con un centinaio di modifiche al giorno o farne una al mese, sempre secondo la sua disponibilità. La sostanziale modularità è anche alla base della commutazione di pacchetto che permette alle informazioni di essere scambiate tra un nodo e l'altro. Attraverso il protocollo TCP/Ip l'informazione di partenza viene suddivisa in tante parti uguali (i pacchetti per l'appunto) e ad ogni pacchetto viene dato l'indirizzo del nodo di arrivo

(l'indirizzo IP). I pacchetti viaggiano separatamente, sfruttando tutto il potenziale del network, spostandosi da un nodo all'altro finché non raggiungono il nodo di arrivo. In corrispondenza del nodo di arrivo i pacchetti vengono ricompattati e l'informazione viene trasmessa al destinatario. Ciò permette di non intasare le linee dirette tra un nodo e l'altro ma di sfruttare tutte le linee momentaneamente inattive per far passare i vari pacchetti. Si sfrutta dunque una proprietà intrinseca dell'architettura di Internet (la struttura a network) massimizzandone i vantaggi.

Il secondo fondamento è la ricombinazione. Anch'esso è contenuto nello stessa infrastruttura: "ogni volta che inviamo una e-mail, gli elaborati algoritmi invisibili del server decidono il percorso che il nostro messaggio seguirà attraverso la rete globale allo scopo di arrivare con il minimo intralcio e la massima velocità. La scelta quantistica forse non svolge alcun ruolo in queste decisioni; piuttosto, a influenzarle sono miliardi di fattori deterministici che interagiscono tra loro. Poiché districare questi fattori è un problema insolubile, queste scelte sono di fatto decisioni dettate dalla libera volontà della rete, e Internet ne prende miliardi ogni giorno." (Kelly 2011, p.319). Per Internet ciò significa ricombinare in continuazione infinite possibilità di percorsi per districarsi nella rete e far arrivare la nostra mail ai nostri colleghi, costruendo continuamente soluzioni nuove basandosi però su soluzioni già utilizzate in precedenza. Invece per quanto riguarda la produzione sociale, questo tipo di fenomeno è forse più visibile nel mondo del software open source e delle librerie che vengono usate per svilupparlo. Le librerie sono degli archivi di funzioni scritte in linguaggio informatico che vengono utilizzate per sviluppare software o ulteriori altre funzioni. La ricombinazione continua di parti di codice informatico scritto da altri e utilizzato per i propri scopi e infine condiviso e ulteriormente sviluppato, è esattamente quello che viene inteso per ricombinazione: tutti i risultati della produzione sociale vengono continuamente ricombinati tra di loro per creare nuovi prodotti che vengono ricombinati a loro volta da altri utenti, e così via. Si noti come in realtà la modularità e la ricombinazione siano anche alla base dello sviluppo biologico della vita (la ricombinazione dei filamenti di DNA durante la riproduzione e la composizione dei filamenti stessi, composte da proteine, composte a loro volta da amminoacidi ricombinati insieme). Questi due fattori, uniti al sostanziale progresso dell'interfaccia uomo macchina e degli strumenti di interazione tra di essi, definiscono la logica tecnica stessa degli strumenti digitali che suggerisce degli usi aperti, collaborativi e partecipativi, e questo non solo nel campo del software.