



# ORGANIZZARE IL LAVORO DI UN TEAM VIRTUALE: COSA SI PUÒ IMPARARE DAI PROGETTI DI CROWDSOURCING

Posted on 16 Giugno 2016 by Bolici Francesco

Category: [Tecnologia ed Innovazione Organizzativa](#)

Come si può organizzare il lavoro di un team virtuale distribuito geograficamente? Nei progetti di sviluppo software *open source*, una soluzione organizzativa efficace si basa sul modello aperto di sovrapposizione delle attività individuali. Caratteristiche e limiti di questo approccio sono discusse attraverso l'analisi di una recente ricerca, sottolineando le possibili applicazioni manageriali.

## Crowdsourcing: organizzare team virtuali di volontari

Che cosa fareste se vi chiedessero di realizzare un prodotto complesso (ad esempio un piano strategico, un software o una ricerca), solamente con l'aiuto di volontari che lavorino al progetto sporadicamente (nel loro tempo libero) e ognuno comodamente da casa? Fino a pochi anni fa questa richiesta avrebbe presentato grandi difficoltà di realizzazione.

Come Wikipedia<sup>[1]</sup>, Kickstarter<sup>[2]</sup>, e lo sviluppo di software open-source hanno dimostrato, ora esistono risposte organizzative efficaci a questa domanda. Lavorare a distanza, avvalendosi del contributo volontario di persone esterne all'organizzazione è una scelta realizzabile, anche grazie all'evoluzione e alla diffusione delle tecnologie internet e web.

Tradizionalmente "collaborare" significava impegnarsi con altri per il raggiungimento di uno scopo all'interno di alcuni limiti alla partecipazione ritenuti intrinseci e quindi inevitabili: da quelli spaziali (dover essere fisicamente nello stesso luogo) a quelli temporali (dover lavorare in sincrono). Negli ultimi anni la tecnologia ha reso queste barriere sempre meno rigide, permettendo di lavorare con persone geograficamente distanti che partecipano al progetto senza vincoli di orario (potendo auto-organizzare il proprio tempo anche in maniera asincrona rispetto agli altri). Inoltre le organizzazioni hanno la possibilità di superare i propri confini e di rivolgersi indistintamente a chiunque abbia una connessione internet e voglia contribuire a un progetto.

Questi ultimi processi organizzativi sono stati definiti "*crowdsourcing*" (Howe, 2006), termine derivante dalla crisi di *crowd* (folla) e *outsourcing* (esternalizzazione). Il *crowdsourcing* (che chiameremo anche *esternalizzazione in rete*) è una modalità di collaborazione online in cui un'organizzazione chiede alla rete - quindi ad ogni persona che ritiene di avere le competenze necessarie e il desiderio di partecipare - di contribuire ad un progetto in maniera volontaria. L'organizzazione ha il vantaggio di poter avvalersi del contributo dei volontari, e questi soddisfano una loro specifica esigenza, che può essere di apprendimento, di autostima, di riconoscimento sociale o anche monetaria.

Nel tempo ci sono stati numerosi esempi di progetti esternalizzati in rete, tra i più famosi Wikipedia, Kickstarter e Linux. Nel campo della ricerca possiamo citare: Quantum Moves, un gioco online in cui gli utenti risolvendo



dei problemi al computer hanno mostrato inconsapevolmente agli scienziati che la mente umana sembra essere in grado di risolvere complessi problemi quantistici in maniera intuitiva (Sørensen et al. 2016); o Galazy Zoo, una piattaforma in cui decine di migliaia di volontari hanno classificato autonomamente delle galassie attraverso milioni di foto scattate dai telescopi. Anche molte aziende che operano in settori tradizionali hanno testato modelli di *crowdsourcing*: per esempio Lego Ideas permette a chiunque di proporre le proprie idee per nuovi prodotti, e in caso di selezione positiva, questi saranno fabbricati e commercializzati dall'azienda; o il progetto Fiat Mio, il primo caso di *concept car* progettata attraverso un processo aperto e collaborativo che ha coinvolto oltre centosettantamila partecipanti di centosessanta nazionalità per oltre quindici mesi.

Tra i fenomeni più importanti e diffusi di esternalizzazione in rete vi sono i progetti di sviluppo software open source (OS). In questi progetti il codice sorgente [3] è pubblico e accessibile a tutti, così che i potenziali programmatori (spesso volontari distribuiti geograficamente) possano apportare delle modifiche e ridistribuirlo a chiunque altro (nel rispetto di alcune condizioni specifiche previste nella licenza originale). La comunità OS è particolarmente attiva e comprende decine di migliaia di progetti, tra cui alcuni molto conosciuti e diffusi come Linux, OpenOffice, Firefox e Apache.

Poiché i progetti OS sono tra i primi e più riusciti esempi di organizzazione del lavoro distribuito (coinvolgendo un ampio numero di persone per la realizzazione di prodotti complessi), sono spesso presi come riferimento dai manager per organizzare il lavoro dei team virtuali all'interno della propria azienda. Nei prossimi paragrafi descriveremo alcune delle caratteristiche dei progetti OS e discuteremo alcune indicazioni che potrebbero essere utili all'imprenditore o manager che voglia creare e gestire un team distribuito.

## Come si collabora in un team distribuito? Sovrapponendo le attività su più livelli stratificati

Il gruppo di ricerca della Syracuse University, specializzato in team distribuiti, ha recentemente pubblicato un'interessante ricerca sui meccanismi di collaborazione nei progetti di sviluppo OS (Howison e Crowston, 2014).

I ricercatori spiegano che un importante fattore di successo per i progetti OS consiste nel particolare modello di divisione del lavoro che emerge intorno al codice e che definiscono come un processo aperto di sovrapposizione delle attività (*open superposition*).

Il processo di sovrapposizione aperta ruota intorno alle caratteristiche proprie del lavoro da realizzare (il software nel caso OS) che deve essere sempre accessibile per ogni membro del team. Ogni volta che un programmatore volontario decide di contribuire al progetto, la parte di codice da lui sviluppata è sovrapposta - ovvero messa sopra - al codice già esistente. Il lavoro procede quindi attraverso piccoli task, ognuno dei quali ha una sua applicazione e scopo intrinseco. Ogni contributo, stratificato sul codice scritto in precedenza, ha un valore individuale, perché permette di risolvere uno specifico problema, e simultaneamente ha un valore collettivo perché crea (insieme ad altri moduli) la base necessaria per la produzione del prossimo livello di software. Dal punto di vista concettuale il processo appena descritto è simile alla realizzazione di un modellino con le costruzioni, in cui aggiungendo un mattoncino sopra l'altro ottengo il doppio scopo di i. modificare la struttura esistente rendendola più simile a quella che desidero e allo stesso tempo di ii. creare le condizioni necessarie per aggiungervi sopra un ulteriore livello.



Come vedremo in seguito, questo modello ha almeno due vantaggi: a livello individuale favorisce la motivazione di ogni programmatore a contribuire al progetto, mentre a livello collettivo agisce sorprendentemente come meccanismo intrinseco di coordinamento tra le diverse attività. Inoltre, essendo un modello di organizzazione del lavoro piuttosto lasco (perché non richiede necessariamente una forte interazione tra i partecipanti al progetto), la sovrapposizione aperta tende ad incoraggiare una più ampia esplorazione nello spazio delle possibili soluzioni, favorendo l'emergere di una più estesa varietà di idee (Shore et al. 2015).

## Lavorare da soli o rimandare l'attività

La ricerca ha mostrato due schemi che si ripetono nei progetti OS: 1. la maggior frequenza delle attività individuali rispetto a quelle collaborative e 2. la tendenza a rimandare al futuro un compito troppo complesso per un singolo individuo, invece di ridisegnarlo come un lavoro in gruppo. I dati dimostrano che la maggior parte del codice all'interno di un progetto OS è realizzato solamente grazie al lavoro individuale, mentre relativamente pochi moduli sono creati in collaborazione tra più persone. Il secondo punto riguarda le attività che, essendo troppo ampie o complesse per un singolo programmatore, sono spesso rimandate nel tempo invece che organizzate e gestite in maniera collettiva. In pratica, il singolo programmatore preferisce attendere il momento in cui l'attività sarà più semplice da svolgere da solo (perché il codice nel frattempo si è evoluto e le difficoltà specifiche diminuite), piuttosto che dedicare tempo e risorse a convincere e coordinare un team di persone che lo supporti per completare il lavoro.

Questi risultati riguardano direttamente due fattori: uno legato alla motivazione individuale a partecipare al progetto e l'altro ai meccanismi di coordinamento del progetto stesso. Per discutere della motivazione individuale, consideriamo un volontario nel momento in cui debba decidere se lavorare o meno ad una specifica attività del progetto. Una possibilità è che il programmatore sia spinto principalmente dall'opportunità di imparare, e che la sfida di sviluppare quella specifica parte di codice possa soddisfare questo desiderio. In questo caso il volontario si dedicherà all'attività scelta a prescindere dalla possibilità che il codice scritto sia effettivamente utile al progetto. Un caso diverso è quando la motivazione sia legata alla possibilità di aumentare la propria reputazione nella comunità di riferimento o semplicemente al piacere di sviluppare una nuova parte di software utile a sé e gli altri. La sovrapposizione aperta consente di gestire e supportare tutte queste motivazioni individuali, perché lascia al singolo sviluppatore l'autonomia decisionale di scegliere la propria attività di lavoro.

La capacità di attrarre la partecipazione individuale è la base di ogni progetto distribuito aperto, ma non è tutto. Sebbene sviluppatori altamente motivati e autonomi possano produrre anche da soli parti di codice potenzialmente utili, un progetto di sviluppo OS ha bisogno di mettere insieme il lavoro di tutti in un sistema coerente. Lavorare in maniera produttiva vuol quindi dire lavorare con gli altri in modo coordinato. Il coordinamento può essere definito come la gestione delle dipendenze tra le attività (Malone e Crowston 1994; Malone et al. 1999). Il processo di sovrapposizione delle attività risolve intrinsecamente il problema del coordinamento perché uno specifico modulo potrà essere realizzato solo se la struttura su cui si basa (il livello di codice sottostante) è già stata prodotta e resa pubblica. La singola attività, che è motivata individualmente in maniera autonoma (perché ogni modulo del codice assolve una specifica funzione), produce allo stesso tempo un risultato (codice) intorno al quale altri potenziali programmatori possono coordinare il proprio lavoro futuro.



In sintesi, la sovrapposizione del lavoro individuale è la modalità principale di organizzazione del lavoro in progetti OS perché questa organizzazione presenta simultaneamente le più basse dipendenze tra attività (minor coordinamento richiesto) e la situazione motivazionale più semplice da gestire. In particolare, la sovrapposizione del lavoro individuale porta ad un'alta spinta motivazionale perché aumenta l'autonomia e la competenza individuale, senza eliminare la possibilità dell'individuo di relazionarsi con gli altri e con il progetto nel suo complesso. Il lavoro che non può rientra in questo modello può essere gestito attraverso attività condivise tra più persone, se l'attività ha buone probabilità di venir completata e la perdita di autonomia è bilanciata da un maggiore livello relazionale. Se queste condizioni non sono soddisfatte, come spesso succede, il lavoro è rimandato fino anche altre attività completate lo renderanno realizzabile attraverso la sovrapposizione del lavoro individuale.

## Implicazioni manageriali: quali consigli per organizzare il lavoro di un team distribuito?

Per comprendere a pieno i vantaggi di un progetto distribuito che adotti un modello di sovrapposizione delle attività, proviamo a immaginare le difficoltà che incontrerebbe un manager che seguisse un approccio tradizionale nella gestione di un progetto OS. Dovrebbe innanzitutto identificare i risultati e il disegno della sequenza di attività da implementare, e quindi dovrebbe trovare programmatori in grado e disponibili a realizzare tale sequenza rispettando i tempi stabiliti per ogni attività. La gestione di queste meta-attività rappresenta il costo del progetto. Come fanno i progetti OS a realizzare queste attività senza sostenere direttamente questi costi, in altre parole senza avere un manager che si occupi a tempo pieno di queste attività?

La ricerca della scuola di Syracuse sostiene che i progetti OS di successo svolgano queste attività gestionali ad un costo molto basso grazie alla disponibilità e trasparenza del software che producono. Una caratteristica essenziale dei progetti OS è che il software stesso (inteso come oggetto del lavoro) sia sempre disponibile e accessibile a tutti a costo zero. Poiché in ogni momento lo stato di avanzamento del lavoro è visibile a tutti, ogni volontario può costantemente "visualizzare la prossima attività sulla base di quello che è già stato sviluppato" (p. 41). La possibilità di confrontarsi costantemente con lo stato di tutte le attività del progetto significa che le future attività identificate dagli sviluppatori avranno una maggiore probabilità di essere appropriate dal punto di vista sequenziale. Un'architettura software composta da piccole parti di codice trasparente e accessibile rende più facile generare nuove idee, far sviluppare la motivazione individuale a partecipare e condividere la conoscenza tra gli sviluppatori. Sviluppare in piccole parti di codice genera un prodotto riusabile, di alta qualità, facile da capire e che consente di rimandare al futuro un'attività non realizzabile individualmente nel presente.

Perché un processo come quello descritto sia realizzabile devono verificarsi alcune condizioni: 1. l'oggetto del lavoro deve avere alcune specifiche caratteristiche; 2. il modello di condivisione dei contributi deve essere rigidamente aperto; 3. non devono sussistere stretti limiti di natura temporale.

## Caratteristiche dell'oggetto del lavoro

Gli attributi dell'oggetto del lavoro sono cruciali perché si possa adottare un modello di collaborazione basato



sulla sovrapposizione aperta. In particolare, oltre ad essere trasparente e accessibile per tutti i membri del progetto, il prodotto del lavoro: 1. deve poter essere sviluppato in strati o livelli; 2. deve avere bassi costi di realizzazione; 3. deve avere bassi costi di distribuzione. La possibilità di stratificare vuol dire poter sviluppare il lavoro collettivo come una sequenza di strati, ognuno dei quali abbia un suo specifico scopo che non dipende da attività future. Avere bassi costi di realizzazione vuol dire che i costi necessari per passare dalla fase di progettazione a quella di realizzazione sono limitati. Se fosse costoso il processo di aggiungere un ulteriore strato sul lavoro già svolto, diventerebbe meno conveniente modificare spesso il lavoro. Infine, bassi costi di distribuzione sono necessari per far crescere in fretta la comunità che usa il prodotto e che potenzialmente può avere un interesse a parteciparvi. Inoltre, costi bassi permettono ai nuovi strati di prodotto di essere distribuiti velocemente e capillarmente.

## Modello di condivisione rigidamente aperto

Un modello di condivisione rigidamente aperto descrive una situazione tecnica, legale ed etica necessaria perché la collaborazione tramite sovrapposizione possa avvenire. I volontari devono essere certi che ogni contributo aggiunto al progetto sia e sarà sempre nella piena disponibilità del progetto stesso, senza la possibilità per chi lo abbia creato di negarne l'uso (anche in futuro). Nel caso opposto, infatti, tutti gli strati che sono stati costruiti su quel contributo sarebbero a rischio, perché si basano sulla sua presenza e sulle sue caratteristiche.

## Bassa pressione temporale

La collaborazione tramite sovrapposizione richiede molto tempo per generare un prodotto utile. Occorre molto tempo perché il prodotto si diffonda all'interno di una comunità in modo da raccogliere la necessaria attenzione, ne serve altro perché i primi volontari decidano di contribuire al suo miglioramento, e altro ancora perché il lavoro rimandato possa trovare le condizioni adatte per essere finalmente affrontato. L'imposizione di rigide scadenze per il completamento di un'attività o di un modulo sarebbe probabilmente controproducente nella logica di garantire una più ampia esplorazione delle possibili soluzioni per quello specifico problema. Questo perché da una parte i volontari sarebbero scoraggiati dal partecipare e dall'altra sarebbe più difficile rimandare l'esecuzione di quell'attività, con la conseguenza di scegliere la prima soluzione accettabile pur di rispettare la scadenza.

## Conclusioni

Organizzare il lavoro di un team distribuito è un'attività complessa. Sempre più aziende sono interessate a sviluppare una comunità di volontari sia all'interno dei confini organizzativi –si pensi a tutti i tentativi di coinvolgere i propri dipendenti in una intranet aziendale in cui scambiare informazioni e conoscenze-, sia all'esterno per coinvolgere un ampio numero di volontari e incrementare la varietà di possibili soluzioni per alcuni specifici problemi. In quest'ottica, manager e aziende potrebbero avere interesse a confrontarsi con le soluzioni organizzative sviluppatesi nei progetti OS. Infatti molte delle difficoltà che si incontrano nel lavoro distribuito sono già state risolte, anche se in condizioni specifiche, all'interno delle comunità di sviluppo software.



Abbiamo discusso come alcuni ricercatori abbiano individuato nel modello di sovrapposizione aperto un elemento cruciale per raggiungere il successo nei progetti OS. La sovrapposizione, insieme ad un'ampia distribuzione del prodotto e un modello di condivisione che sia irrevocabile nel tempo sono gli elementi da pianificare per poter aumentare le possibilità di successo di un team distribuito.

Esistono naturalmente anche dei limiti all'applicabilità di questo modello, primo tra tutti l'esigenza di terminare il progetto in tempi rapidi e prestabiliti. La collaborazione tramite sovrapposizione può essere molto lenta ed incerta, soprattutto se paragonata a una gestione manageriale tradizionale, che tende a governare e organizzare risorse certe, spesso costose, ma che possono essere dirette gerarchicamente. Tuttavia il modello della sovrapposizione aperta può funzionare per quelle aziende che sono pronte a sostenere una cultura esplorativa basata sul concetto di autonomia del singolo dipendente (es. la politica del 20% del "tempo libero" di Google). Inoltre la collaborazione tramite sovrapposizione può rappresentare una vantaggiosa opzione organizzativa nel caso il problema da risolvere riguardi l'esternalizzazione delle attività ad una forza lavoro sconosciuta e non direttamente controllabile.

## Bibliografia

Estellés-Arolas, E., & González-Ladrón-de-Guevara, F. (2012). Towards an integrated crowdsourcing definition. *Journal of Information science*, 38(2), 189-200.

Howe, J. (2006). The rise of crowdsourcing. *Wired magazine*, 14(6), 1-4.

Howison, J., & Crowston, K. (2014). Collaboration Through Open Superposition: A Theory of the Open Source Way. *MIS Quarterly*, 38(1), 29-50.

Leimeister, J. M., Huber, M., Bretschneider, U., & Krcmar, H. (2009). Leveraging crowdsourcing: activation-supporting components for IT-based ideas competition. *Journal of management information systems*, 26(1), 197-224.

Malone, T. W., & Crowston, K. (1994). The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1), 87-119.

Malone, T. W., Crowston, K., Lee, J., Pentland, B., Dellarocas, C., Wyner, G., & Klein, M. (1999). Tools for inventing organizations: Toward a handbook of organizational processes. *Management Science*, 45(3), 425-443.

Shore, J., Bernstein, E., & Lazer, D. (2015). Facts and figuring: An experimental investigation of network structure and performance in information and solution spaces. *Organization Science*, 26(5), 1432-1446.

Sørensen, J. J. W., Pedersen, M. K., Munch, M., Jensen, J. H., Planke, T., Gajdacz, M. G. A. M., & Sherson, J. F. (2016). Exploring the Quantum Speed Limit with Computer Games. *Nature* (532), 210-213.

[1] Wikipedia è un'enciclopedia online collaborativa e gratuita [wikipedia.org]. Per la sola versione italiana, più di 8.000 volontari hanno contribuito con almeno una modifica all'enciclopedia nell'ultimo mese (fonte: [https://it.wikipedia.org/wiki/Pagina\\_principale](https://it.wikipedia.org/wiki/Pagina_principale), Aprile 2016)

[2] Kickstarter è uno dei principali siti per il finanziamento collettivo (*crowdfunding*) [kickstarter.com]. Gli utenti propongono dei progetti creativi e tutti (dal basso) possono decidere volontariamente se sostenerlo,



collaborando in questa maniera ad un processo di micro-finanziamento.

[3] Il codice sorgente è il testo di un programma, scritto in uno specifico linguaggio di programmazione, che potrà poi essere eseguito.