

# Linee guida RUP«/XP: Programmazione a coppia

Robert C. Martin  
Object Mentor, Inc.

---

White Paper del software Rational

TP 158, 3/01

**Rational®**  
the software development company

## Indice

Panoramica ...	1
Lavoro di coppia, una breve descrizione ...	1
Il caso del lavoro di coppia...	1
La pratica ...	1
Lavoro di coppia...	1
Scambio di coppie ...	2
Collective ownership ...	2
Compensazione e collaborazione...	2
Cosa può fare lo sviluppatore singolo? ...	2
Ad alcuni non piace il lavoro di coppia...	2
Mobili, attrezzature e logistica ...	3
Disposizione del monitor e della tastiera ...	3
Open space ...	3
Angoli interni ...	3
Problemi e questioni ...	3
Produttività del lavoro di coppia ...	3
Dispute tra partner di una coppia ...	3
Specialisti ...	3
Rumore ...	3
Cowboy ...	3
Impedimenti fisici e di stile ...	4
In che modo il team può pianificare il lavoro di coppia? ...	4
Conclusioni...	4
Riferimenti ...	4

## Panoramica

---

### Lavoro di coppia, una breve descrizione

La programmazione a coppia è una tecnica con cui il software di un progetto viene scritto da coppie di programmatori, ciascuna delle quali lavora insieme da un'unica postazione. Un membro della coppia dirige la postazione mentre l'altro segue, controllando accuratamente il codice che si sta producendo. Il primo pensa in modo tattico, attento alla linea di codice che sta scrivendo. Il secondo si occupa di convalidare la sintassi e pensa in modo strategico all'intero programma. I ruoli vengono frequentemente scambiati, in modo da riuscire ad ottenere una maggiore velocità e precisione della compilazione del codice. Inoltre il codice è conosciuto a fondo da almeno due sviluppatori.

### Il caso del lavoro di coppia

Si prenda in considerazione una comune sessione di revisione del codice. Un modulo che richiedeva per lo sviluppo otto ore a persona viene revisionato in un'ora da otto persone. Il risultato netto è che sul modulo vengono impiegate 16 persone-ore. Tuttavia i revisori non hanno la possibilità di impiegare il tempo necessario ad acquisire familiarità con il codice e la revisione risulta alquanto superficiale. Un solo sviluppatore ha più familiarità, ma forse troppa per individuare le imperfezioni.

Si faccia un paragone con la programmazione a coppia. Se il modulo richiede per lo sviluppo otto ore a coppia, verrà impiegato il totale di 16 persone-ore. Tuttavia, in questo caso due sviluppatori avranno una conoscenza approfondita del codice. I difetti sfuggiti ad uno sviluppatore saranno visibili per l'altro.

Il caso della programmazione a coppia è semplice, ma le ripercussioni sono minime e di vasta portata. La programmazione a coppia è semplicemente un modo più efficace per scrivere e revisionare i codici. La presenza di due persone con una profonda conoscenza del modulo fa sì che nel codice vengano scritti meno errori. Il codice avrà una migliore struttura e due persone ne avranno un'intima conoscenza. Se fossero questi gli unici vantaggi sarebbero già sufficienti, ma ve ne sono degli altri.

Le coppie hanno più coraggio: quello che un singolo programmatore non farebbe, una coppia ha il coraggio di tentare e la capacità di valutare.

Il lavoro di coppia incoraggia il team di lavoro: poiché i moduli non sono scritti da una sola persona, il codice diventa di proprietà del team, piuttosto che di uno sviluppatore in particolare.

Il lavoro di coppia incoraggia la diffusione della conoscenza: più sviluppatori lavorano in coppia, più la conoscenza del sistema si diffonderà nell'intero team. Il risultato è un team i cui membri hanno familiarità con tutto il sistema piuttosto che conoscerne solo una parte specifica.

Il lavoro di coppia favorisce la produttività: una persona che programma da sola attraversa periodi di piena energia seguiti da periodi di relativa inattività. Le coppie si compensano. Quando un membro è stanco, i ruoli vengono invertiti. Si riesce così a mantenere alta l'intensità più a lungo di quanto possa fare una singola persona.

Il lavoro di coppia è divertente: lavorare a fianco di un altro sviluppatore è educativo, stimolante e anche divertente. Il lavoro di coppia aumenta la soddisfazione del lavoro e solleva il morale.

---

## La pratica

### Accoppiamento

L'accoppiamento inizia quando lo sviluppatore responsabile di un compito richiede l'aiuto di qualcun'altro. La regola è essere sempre disponibili. Ciò non significa interrompere immediatamente il lavoro che si sta svolgendo, ma negoziare il momento in cui è possibile offrire l'aiuto chiesto e il momento in cui si avrà bisogno di ricevere aiuto.

I partner della coppia non si assumono la responsabilità del compito, che resta invece del proprietario. Né si impegnano a stare con il proprietario fino al completamento del compito. Il partner si impegna solo a dare aiuto.

Un membro della coppia diventa la guida e l'altro si occupa di controllare. La guida scrive il codice, esegue il compilatore, i test di unità e così via. L'osservatore esamina ogni battitura di tasti, ogni comando, ogni risultato di test e offre aiuto e suggerimenti. Entrambe le parti sono impegnate per tutto il tempo.

Potrebbe accadere che la guida sappia meglio cosa fare e l'osservatore si limiterà a seguire. Altre volte l'osservatore detterà alla guida cosa fare. Può anche capitare che la guida si scoraggi e passi la tastiera all'osservatore, scambiando così i ruoli, o che l'osservatore chiedi la tastiera e di alternare i ruoli. Tutto ciò può avvenire più volte durante una sessione di coppia.

## Scambio di coppie

I partner di una coppia non lavorano insieme a lungo termine. Una tipica sessione di coppia dura in genere mezza giornata. Entrambi i partner possono ritirarsi per qualunque motivo. In questo caso, il proprietario del compito deve provvedere ad un altro partner. Può essere questo il momento in cui il proprietario restituisce il favore a qualcuno con cui ha fatto coppia nella settimana passata. D'altro canto, potrebbe richiedere una persona con l'esperienza adatta a risolvere un problema particolarmente sgradevole.

Tale scambio di coppie consente di diffondere la conoscenza del sistema all'intero team di sviluppo. In breve tempo, ogni membro avrà avuto la possibilità di lavorare su quasi ogni parte del sistema, riducendo fortemente la sensibilità del progetto al turnover e rendendo ogni programmatore più fiducioso nell'approccio al sistema.

## Collective ownership

Poiché ognuno lavora su tutti i diversi moduli, nessuno ne possiede uno in particolare. Ciò significa che la responsabilità del sistema non è suddivisa modulo per modulo. Al contrario, tutta la squadra è collettivamente responsabile dell'intero sistema. Ogni membro può eseguire il check out e modificare ogni modulo nel sistema per qualunque motivo. Se una coppia esegue una modifica sul modulo X determinando il fallimento del test di unità del modulo Y, la coppia riparerà il modulo Y.

## Compensazione e collaborazione

La programmazione in coppia è una forma di comunicazione molto intensa. Il dialogo verbale è di solito scarso e un osservatore esterno potrebbe avere problemi a comprenderlo. Si potrebbe sentire la coppia pronunciare parole quali "punto e virgola" o "chiusa parentesi", o semplicemente suoni meno articolati che indicano l'accordo o il disaccordo dei programmatori con ciò che appare sullo schermo. I due sono talmente concentrati sul codice che buona parte della comunicazione è non verbale. Il linguaggio corporeo gioca anche un ruolo importante. Un membro della coppia capisce se il partner è d'accordo con il codice senza

bisogno di parole. Una smorfia, una smania nervosa, un sospiro contribuiscono ad aumentare la larghezza di banda della comunicazione tra i partner. Può capitare che un partner afferri il mouse mentre l'altro opera sulla tastiera. Il primo controlla la posizione all'interno del modulo in cui il lavoro verrà svolto. Il secondo controlla il contenuto alterato o aggiunto in quella posizione. Altre volte mentre un partner sta scrivendo, l'altro potrebbe prevedere l'arrivo di una chiamata di funzione e aprire i documenti API alla pagina giusta proprio quando il collega ha bisogno della specifica.

Quando un partner è stanco, l'altro può assumere la guida permettendo al collega di riposarsi nel ruolo di osservatore. Altre volte entrambi i partner avranno molta energia e si scambieranno tastiera e mouse frequentemente.

In breve, esistono poche regole e poche procedure. L'unico vincolo è che entrambi devono impegnarsi e mantenere intensa la comunicazione. Una coppia in cui un partner scrive e l'altro guarda fuori dalla finestra non è propriamente una coppia.

## Cosa può fare lo sviluppatore singolo?

Non è possibile lavorare in coppia in ogni caso. Alcuni progetti, ad esempio quelli che adottano il processo eXtreme Programming (XP) (vedere il riferimento [1]), seguono la regola secondo cui le coppie devono produrre tutto il codice di produzione. In questo caso, se non si è in coppia è possibile controllare la posta, leggere di una nuova tecnica o un nuovo API, leggere codici con cui non si ha familiarità o parlare con gli stakeholder sull'iterazione attuale o sui progetti futuri. In realtà c'è sempre qualcosa di produttivo che uno sviluppatore può fare in quelle poche ore in cui non si ha un partner.

Alcuni progetti sono meno rigidi riguardo le coppie, qualcuno permette ai singoli sviluppatori di scrivere test, altri di scrivere classi o interfacce astratte, altri ancora consentono agli sviluppatori di decidere quando fare coppia. Tuttavia una cosa è chiara, alcuni studi hanno rivelato che l'occorrenza di errori diminuisce notevolmente quando si lavora in coppia.

## Ad alcuni non piace lavorare in coppia

Alcune persone non condividono il concetto di lavorare in coppia. Secondo la nostra esperienza, coloro che effettivamente lo provano per una settimana o più abbandonano le diffidenze e iniziano a praticare e trovare utile il lavoro in coppia. Pochi continuano a non preferirlo. Per molti dunque è solo questione di provare e abituarsi. Per coloro che, pur avendo effettivamente provato, continuano a non trovare interessante questa pratica, il team dovrà trovare qualcosa di adeguato da fare.

## Mobilia, attrezzature e logistica

---

### Disposizione del monitor e della tastiera

La disposizione dei mobili è fondamentale per il successo del lavoro di coppia. Non si lavorerà bene se il partner non potrà sedersi accanto e scambiare rapidamente la tastiera. La regola è essere in grado di scambiarsi tastiera e mouse senza cambiare posto.

La migliore sistemazione è spesso un tavolo lungo e stretto. Posizionare il monitor al centro e le due sedie di fronte. Sedersi avendo il monitor nel mezzo. Assicurarsi di poter far scorrere tastiera e mouse facilmente. Assicurarsi anche che mentre si ha la tastiera si stia seduti comodamente e dritti. Sistemare il monitor in modo che sia visibile ad entrambi i partner senza doverlo girare.

### Open space

Per facilitare lo scambio di coppie partner è consigliabile lavorare in un ambiente open space. Posizionare diverse stazioni di coppia in una singola stanza. Utilizzare sedie con rotelle e pavimenti di linoleum o a mattonelle. Sistemare le postazioni di lavoro di modo che le coppie siano una di fronte all'altra. L'obiettivo è aumentare la comunicazione. A volte le comunicazioni più importanti sono quelle più spontanee, dunque si dovrebbero aumentare le possibilità di occorrenza di queste comunicazioni.

### Angoli interni

In molti scompartimenti oggi le postazioni di lavoro sono sistemate negli angoli. Lo sviluppatore siede rivolto verso un angolo della cabina con un monitor di fronte. Se ciò è adatto per un lavoro individuale, è quasi impossibile lavorare bene in coppia in un ambiente simile. Se si ha una sistemazione di questo tipo, organizzare alcune postazioni per coppia altrove, magari in una sala conferenze. Lavorare in coppia sul tavolo di una sala conferenze con un computer portatile risulta spesso efficace.

## Problemi e questioni

---

### Produttività del lavoro di coppia

Vi sono ragioni di credere che due persone che lavorano insieme su un compito impiegheranno il doppio di persona-ore rispetto ad una persona che lavora sullo stesso compito. Per quanto ciò sia probabile, non è questo il caso. Alcuni studi indipendenti (vedere il riferimento [2]) hanno dimostrato che la carenza di produttività non riguarda il lavoro in coppia. Gli stessi studi mostrano che il lavoro in coppia in sostanza diminuisce il tasso di errori e la misura del codice, con una crescita dell'entusiasmo lavorativo.

### Dispute tra partner di coppia

Il proprietario del compito ha l'ultima parola in tutte le dispute di progettazione, ma il modo migliore per risolvere una controversia è di provare entrambe le idee e scegliere quella che funziona meglio.

### Specialisti

Per convenzione, gli sviluppatori specializzati in un'area particolare, come il databases o GUI, dovrebbero investire il proprio impegno unicamente in quelle aree. Tuttavia, in un ambiente di programmazione di coppia, tali specialisti diventano una guida per gli altri che non conoscono quella specializzazione. Gli sviluppatori possono impegnarsi in un compito al di fuori del proprio ambito e richiedere l'aiuto di specialisti. In questo modo, la conoscenza di specializzazioni tende a diffondersi nel team del progetto, riducendo la sensibilità del progetto al turnover.

### Rumorosità

Una coppia potrebbe fare chiasso nel programmare. Un open space pieno di coppie aiuterà a mantenere un continuo ma lieve brusio. Alcuni potrebbero ritenere che questo rumore sia noioso e che distragga. Non è stato dimostrato che questo sia un problema significativo. Se si ritiene di essere distratti dal rumore è possibile spostarsi nella sala conferenze per un po'.

### Cowboy

Molti team ritengono, con orgoglio, di avere uno o più codificatori cowboy, ovvero quei programmatori che terminano il lavoro prima degli altri, non riescono a lavorare in coppia e non consentono ad altri (che forse non vi riuscirebbero) di leggere

il proprio codice. La miglior cosa da fare con questi sviluppatori è di allontanarli dal progetto e assegnargli un ruolo in cui non debbano scrivere un codice di produzione. Potrebbero forse scrivere tool di breve durata o eseguire dei test di tortura maniacale.

### Impedimenti fisici e di stile

Alcuni utilizzano tastiere QWERTY, altri preferiscono quelle DVORAK, altri ancora hanno bisogno di speciali tastiere, mouse, schermi, interruttori a pedale e così via. Alcuni non riescono a programmare senza cuffie o musica ad alto volume, altri devono circondarsi di pacchetti vuoti di biscotti. Alcuni preferiscono Emacs o VI, altri ancora vorrebbero lavorare con WordPad.

Si potrebbero nominare infiniti impedimenti, ma ad ognuno si può trovare una soluzione pensandoci un po' e avendo predisposizione al compromesso. Una squadra che si lascia influenzare da tali cose è una squadra che non riuscirà in nessun tentativo.

### In che modo il team può pianificare il lavoro di coppia?

Non è consigliabile assegnare compiti alle coppie, ma piuttosto ogni sviluppatore deve essere responsabile di una serie di compiti. È preferibile che le coppie si formino in modo informale. Ogni sviluppatore, seguendo le proprie responsabilità, chiede ad altri sviluppatori un breve aiuto. Il proprietario del compito mantiene la propria responsabilità e il proprio compito. Le coppie fungono solo da aiuto. Ogni sviluppatore deve tenere in conto il tempo che si impiega a lavorare in coppia nel fornire stime per i compiti.

## Conclusioni

---

La programmazione a coppie è un'alternativa valida e certificata per le revisioni di codice. Rappresenta inoltre un modo essenzialmente diverso di scrivere software. I vantaggi non si limitano a produttività e qualità, ma influenzano la solidità e il morale di un team.

## Riferimenti

---

[1] eXtreme Programming eXplained, Kent Beck, Addison Wesley, 2000.

[2] Strengthening the Case for Pair Programming, Laurie Williams, University of Utah, July/Aug 2000 IEEE Software.

# Rational®

the software development company

Sedi principali:

Rational Software  
18880 Homestead Road  
Cupertino, CA 95014  
Tel: (408) 863-9900

Rational Software  
20 Maguire Road  
Lexington, MA 02421  
Tel: (781) 676-2400

Numero verde: (800) 728-1212

E-mail: [info@rational.com](mailto:info@rational.com)

Sito Web: [www.rational.com](http://www.rational.com)

Sito internazionale: [www.rational.com/worldwide](http://www.rational.com/worldwide)

Rational, il logo Rational e Rational Unified Process sono marchi registrati di proprietà di Rational Software Corporation negli Stati Uniti e/o in altri Paesi. Microsoft, Microsoft Windows, Microsoft Visual Studio, Microsoft Word, Microsoft Project, Visual C++, e Visual Basic sono marchi di fabbrica o marchi registrati di proprietà di Microsoft Corporation. Tutti gli altri nomi vengono utilizzati solo per fini di identificazione e sono marchi o marchi registrati delle rispettive società. TUTTI I DIRITTI RISERVATI. Made in USA

© Copyright 2002 Rational Software Corporation.

Il contenuto può essere soggetto a modifiche senza preavviso.