Introduzione

Prima di iniziare, ci sono due questioni che dobbiamo affrontare per assicurarci che voi, miei gentili lettori, comprendiate il quadro di riferimento in cui si colloca questo libro.

Sul termine "craftsmanship"

L'inizio del XXI secolo è caratterizzato da alcune controversie sulla lingua, e anche noi dell'industria del software abbiamo le nostre controversie. Un termine piuttosto discusso è proprio *craftsman*, artigiano.
Ho riflettuto un po' su questo problema, ne ho parlato con molte persone che avevano opinioni differenti e sono giunto alla conclusione che non esiste un termine migliore, nel contesto di questo libro.
L'*artigiano* fa pensare a una persona profondamente competente e abile in una particolare attività, qualcuno che conosce alla perfezione i propri attrezzi e il proprio mestiere, che è orgoglioso del proprio lavoro e di cui ci si può fidare: si comporterà con dignità e professionalità spinto dalla propria vocazione.
Può darsi che alcuni di voi non saranno d'accordo con questa decisione, e li capisco. Spero solo che non interpretiate quel termine come un desiderio di essere in alcun modo esclusivo, perché non è affatto il mio intento.

Sull'unica vera via

Leggendo *Clean Craftsmanship*, potreste avere la sensazione che sia l'*unica vera via* della programmazione. Può essere così *per me*, ma non necessariamente per voi. Vi offro questo libro come un esempio della *mia via*. Ovviamente voi dovrete scegliere la vostra.
Ma avremmo davvero bisogno di un'unica vera via? Non lo so. Forse. Come leggerete in queste pagine, cresce la pressione per una definizione rigorosa di che cosa sia un professionista dello sviluppo software. Potremmo essere in grado di cavarcela seguendo più vie, a seconda della criticità del software da creare. Ma, come leggerete più avanti, non sempre è così facile separare il software critico da quello non critico.
Di una cosa sono certo. I tempi dei "Giudici" dell'Antico Testamento sono finiti. Non è più sufficiente che un programmatore faccia ciò che sembra giusto ai suoi occhi.

È necessario che agisca eticamente, con disciplina, seguendo degli standard. La decisione che abbiamo davanti come programmatori è se questi tre elementi li definiremo noi o ce li faremo imporre da chi non ci conosce.

Introduzione al libro

Questo libro è rivolto ai programmatori e ai manager. Ma in un certo senso, questo libro è scritto per tutta la società umana. Perché noi programmatori ci siamo trovati, inavvertitamente, a essere il fulcro stesso di quella società.

Per voi stessi

Se siete programmatori e avete ormai anni di esperienza sulle spalle, probabilmente conoscete la soddisfazione di riuscire a produrre un software perfettamente funzionante. Si prova un certo orgoglio nell'aver raggiunto tale traguardo. C'è di che essere orgogliosi nell'aver rilasciato un software.

Ma siete orgogliosi anche del *modo* in cui lo avete rilasciato? Siete solo fieri di aver finito oppure del lavoro che avete fatto? Siete orgogliosi che il sistema sia finalmente stato implementato o del modo in cui lo avete realizzato?

Quando tornate a casa dopo una dura giornata trascorsa a scrivere codice, vi guardate allo specchio e dite "Oggi ho fatto proprio un buon lavoro" o avete un bisogno disperato di farvi una doccia?

Troppi di noi si sentono sporchi, a fine giornata. Troppi di noi si sentono intrappolati nel produrre un lavoro scadente. Troppi di noi ritengono che la bassa qualità sia già prevista e che sia l'altra faccia dell'alta velocità. Troppi di noi pensano che produttività e qualità siano inversamente proporzionali.

In questo libro, mi sforzo di spazzare via questa mentalità. Questo è un libro sul *lavorare bene*. Un libro sul fare un buon lavoro. Questo è un libro che descrive le discipline e le pratiche che ogni programmatore dovrebbe conoscere per lavorare velocemente, per essere produttivo e per essere orgoglioso di ciò che scrive, ogni singolo giorno.

Per la società

Nel XXI secolo, per la prima volta nella storia dell'umanità la nostra società è diventata dipendente, per la sua sopravvivenza, da una tecnologia che non ha acquisito praticamente nessuna parvenza di disciplina o controllo. Il software ha invaso ogni aspetto della vita moderna, dalla preparazione del caffè mattutino alla scelta dell'intrattenimento serale, dal lavaggio dei panni alla guida dell'auto, dal connetterci su una rete planetaria fino al dividerci, socialmente e politicamente. Non c'è letteralmente nessun aspetto della nostra vita che non sia dominato dal software. Eppure, coloro che creano tale software sono poco più di un gruppo sgangherato di aggiustatutto che hanno solo una vaga idea di quello che stanno facendo.

Se noi programmatori avessimo avuto una maggiore consapevolezza di ciò che facciamo, i risultati delle primarie presidenziali del 2020 nello Iowa sarebbero arrivati così tardi? Sarebbero morte 346 persone nei due incidenti del 737 Max? Il Knight Capital Group

avrebbe perso 460 milioni di dollari in 45 minuti? Avrebbero perso la vita 89 persone negli incidenti dovuti ad accelerazioni non intenzionali di una Toyota?
Ogni cinque anni, nel mondo, il numero di programmatori raddoppia. A tali programmatori viene insegnato molto poco sul loro mestiere. Vengono loro mostrati gli strumenti, poi vengono loro forniti alcuni progetti giocattolo da sviluppare e infine vengono lanciati per entrare a far parte di una forza lavoro in crescita esponenziale, per rispondere a una domanda, anch'essa in crescita esponenziale, di software. Ogni giorno, il castello di carte che chiamiamo software si insinua sempre più in profondità nella nostra infrastruttura, nelle nostre istituzioni, nei nostri governi e nelle nostre vite. E ogni giorno cresce il rischio di una catastrofe.
Di quale catastrofe parlo? Non del crollo della nostra civiltà né dell'improvvisa avaria di tutti i sistemi software insieme. Il castello di carte che sta per crollare non è composto dai sistemi software stessi. Quello che è a rischio è il fragile fondamento della fiducia del pubblico.
Troppi incidenti del 737 Max, incidenti di accelerazione di una Toyota, incidenti Volkswagen denunciati dall'EPA o incidenti alle elezioni primarie dello Iowa: ulteriori casi di errori o gravi illeciti nel software uniti alla mancanza di disciplina, etica e standard saranno al centro di un dibattito pubblico carico di diffidenza e rabbia. Poi seguiranno le regolamentazioni: regole che nessuno di noi desidera, che paralizzeranno la nostra capacità di esplorare ed espandere liberamente l'arte dello sviluppo software, che imporranno gravi restrizioni alla crescita della nostra tecnologia ed economia.
L'obiettivo di questo libro non è quello di fermare la corsa precipitosa verso una sempre maggiore adozione del software, né quello di rallentare il tasso di produzione del software. Tali obiettivi sarebbero solo uno spreco di energie. La nostra società ha bisogno di software e lo otterrà a qualunque costo. Il tentativo di soffocare questo bisogno non fermerà l'incombente catastrofe della fiducia pubblica.
L'obiettivo di questo libro è imprimere nella mente degli sviluppatori e dei loro manager la necessità di disciplina, e insegnare le discipline, gli standard e l'etica più efficaci per massimizzare la capacità di realizzare software solidi, *fault-tolerant* ed efficaci. È solo cambiando il modo in cui noi programmatori lavoriamo, elevando la nostra disciplina, la nostra etica e i nostri standard che il castello di carte può essere rafforzato, per impedire che crolli.

La struttura di questo libro

Questo libro è suddiviso in tre parti, che descrivono altrettanti livelli: le discipline, gli standard e gli aspetti etici.
Le *discipline* rappresentano il livello più basso. Questa parte del libro è pragmatica, tecnica e prescrittiva. I programmatori di tutti i livelli trarranno beneficio dalla sua lettura e comprensione. Nelle pagine di questa parte sono presenti diversi riferimenti a video. Questi video mostrano in tempo reale il ritmo delle discipline di sviluppo e *refactoring* basate su test. Anche le pagine del libro cercano di catturare quel ritmo, ma nulla è più efficace dei video.
Gli *standard* rappresentano il livello intermedio. Questa parte delinea le aspettative del mondo nei confronti della nostra professione ed è dedicata in particolare ai manager, in modo che sappiano che cosa aspettarsi dai programmatori professionisti.

L'*etica* è il livello più elevato. Questa parte descrive il contesto etico della professione di programmatore. Lo fa sotto forma di un giuramento, una serie di promesse, ed è ricca di discussioni storiche e filosofiche. Dovrebbe essere letta sia dai programmatori sia dai loro manager.

Una nota per i manager

Queste pagine contengono una grande quantità di informazioni utili per voi. Ma contengono anche un bel po' di materiale tecnico che probabilmente non vi serve. Il mio consiglio per voi è quello di leggere l'introduzione di ciascun capitolo e di fermarvi quando il contenuto inizia a farsi più tecnico del necessario. A quel punto saltate al capitolo successivo.
Assicuratevi di leggere la Parte II e la Parte III, e assicuratevi di leggere le introduzioni di ciascuna delle cinque discipline.

Video di supporto

Sono disponibili cinque video (in inglese) correlati ad alcune sezioni del Capitolo 2 e del Capitolo 3. Per consultarli visitate l'indirizzo: `http://bit.ly/apo-clean-craft`.

Ringraziamenti

Un grazie ai miei intrepidi revisori: Damon Poole, Eric Crichlow, Heather Kanser, Tim Ottinger, Jeff Langr e Stacia Viscardi. Mi hanno salvato da molti inciampi.
Grazie anche a Julie Phifer, Chris Zahn, Menka Mehta, Carol Lallier e a tutti coloro che in Pearson lavorano instancabilmente per produrre così bene questi libri.
Come sempre, grazie alla mia creativa e talentuosa illustratrice, Jennifer Kohnke. Le sue immagini mi fanno sempre sorridere.
E, naturalmente, grazie alla mia adorabile moglie e alla mia meravigliosa famiglia.