

PROJECT MANAGEMENT

# La gestione efficace dei progetti in contesti di cambiamento

I vantaggi dell'Agile Project Management

**Robert B. Dilts**  
**Guido Granchi**

FRANCOANGELI

## **Am** - La prima collana di management in Italia

Testi advanced, approfonditi e originali, sulle esperienze più innovative in tutte le aree della consulenza manageriale, organizzativa, strategica, di marketing, di comunicazione, per la pubblica amministrazione, il non profit...

I lettori che desiderano informarsi sui libri e le riviste da noi pubblicati possono consultare il nostro sito Internet: [www.francoangeli.it](http://www.francoangeli.it) e iscriversi nella home page al servizio “Informatemi” per ricevere via e.mail le segnalazioni delle novità o scrivere, inviando il loro indirizzo, a “FrancoAngeli, viale Monza 106, 20127 Milano”.

**Robert B. Dilts**  
**Guido Granchi**

# La gestione efficace dei progetti in contesti di cambiamento

**I vantaggi dell'Agile Project Management**



FRANCOANGELI

Isbn e-book: 9788835192916

Copyright © 2026 by FrancoAngeli s.r.l., Milano, Italy.

*L'opera, comprese tutte le sue parti, è tutelata dalla legge sul diritto d'autore.  
Sono riservati i diritti per Text and Data Mining (TDM), AI training e tutte le tecnologie simili.  
L'Utente nel momento in cui effettua il download dell'opera accetta tutte le condizioni della  
licenza d'uso dell'opera previste e comunicate sul sito [www.francoangeli.it](http://www.francoangeli.it)*

Eventuali link attivi presenti nel volume sono forniti dall'autore. L'editore non si assume  
alcuna responsabilità sui link ivi contenuti che rimandano a siti  
non appartenenti a FrancoAngeli.

## Parte 1

### I principi del Project Management Agile

<b>1. I principi e i valori del <i>Framework Scrum</i></b>	pag.	11
1.1. Project Management tradizionale e Agile: un parallelo	»	11
1.1.1. L'analisi dei contesti di sviluppo di un progetto: il Modello <i>Cynefin</i>	»	14
1.1.2. Test di auto-valutazione: “La mia azienda è pronta per il Project Management Agile?”	»	19
1.2. I principi del Manifesto Agile: le origini del <i>Framework Scrum</i>	»	20
1.3. Empirismo, trasparenza, ispezione e adattamento	»	26
1.4. I 5 valori di Scrum	»	28

## Parte 2

### I ruoli del *Framework Scrum*

<b>2. Il Product Owner e lo Scrum Master</b>	»	33
2.1. I ruoli del Project Management tradizionale e quelli dell'Agile: un parallelo	»	33
2.2. Il Product Owner: la “voce del Cliente”	»	36
2.2.1. Le responsabilità del Product Owner	»	37
2.3. Lo Scrum Master: il “facilitatore” del progetto Agile	»	39
2.3.1. Le responsabilità dello Scrum Master	»	42
2.4. Lo Scrum Master come Servant Leader	»	43

<b>3. Le caratteristiche dei team Agili</b>	pag.	46
3.1. Le caratteristiche di un team Agile	»	46
3.2. L'auto organizzazione del team Agile	»	46
3.2.1. Prima lezione dalla Natura: il volo delle oche	»	48
3.2.2. Seconda lezione dalla Natura: le dinamiche degli stormi	»	51
3.3. L'auto gestione e la Collaborazione Generativa del team Agile	»	53
3.3.1. La terza lezione dalla Natura: le formiche	»	54
3.3.2. I principi di auto organizzazione	»	55
3.4. Dalla Collaborazione Generativa alle critiche costruttive	»	56
3.4.1. Come fare critiche costruttive	»	59
3.5. La gestione dei conflitti nei team Agili	»	60
3.5.1. Le 4 Prospettive sul conflitto	»	62

### Parte 3

#### Gli Artefatti e gli Eventi del *Framework Scrum*

<b>4. Gli Artefatti</b>	»	67
4.1. La definizione degli Artefatti e la loro importanza	»	67
4.2. Il <i>Product Backlog</i> e il ruolo del Product Owner	»	69
4.3. Lo <i>Sprint Backlog</i> : gli elementi costitutivi e le responsabilità	»	71
4.4. L'Incremento di prodotto e la definizione di "fatto"	»	72
<b>5. Gli eventi</b>	»	74
5.1. Gli eventi Scrum e il concetto di <i>time boxed</i>	»	74
5.2. La pianificazione dello Sprint: lo <i>Sprint Planning meeting</i>	»	76
5.2.1. Le fasi dello <i>Sprint Planning meeting</i>	»	77
5.3. Lo Sprint: il cuore pulsante del Framework	»	79
5.4. L'importanza del <i>Daily Scrum</i> e la sua conduzione	»	80
5.5. La Revisione e la Retrospektiva dello Sprint	»	81
5.6. Il <i>Product Backlog Refinement meeting</i>	»	84

## Parte 4

### I processi del *Framework Scrum*

<b>6. I processi di inizio del progetto</b>	pag. 89
6.1. L'analisi di fattibilità del progetto: il <i>Business Case</i>	» 89
6.2. Identificare il Product Owner e la creazione della visione del progetto	» 91
6.3. Identificazione dello Scrum Master e creazione del team di progetto	» 94
6.3.1. L'accordo di Collaborazione e Piano di team building	» 95
6.3.2. Le fasi di costituzione del team di progetto	» 99
6.4. Identificazione degli stakeholder di progetto	» 101
6.4.1. La matrice degli stakeholder di progetto e la Qualità del prodotto/servizio	» 102
6.5. La sviluppo delle Epiche e delle Funzionalità del prodotto	» 108
6.5.1. L'analisi delle Funzioni del prodotto	» 110
6.6. La definizione delle priorità all'interno del <i>Product Backlog</i>	» 113
6.7. Il <i>Release Plan</i> : definizione e importanza	» 114
<b>7. La pianificazione e le stime</b>	» 117
7.1. La criticità delle stime all'inizio del progetto: il cono di incertezza	» 117
7.1.2. Cono di incertezza e analisi dei rischi nei progetti agili	» 119
7.2. Dalle Funzionalità alle <i>User Story</i> : la creazione delle Storie Utente	» 120
7.2.1. Le caratteristiche di una <i>User Story</i> di successo: il modello INVEST	» 121
7.3. I criteri di accettazione delle <i>User Story</i> : definizione e importanza	» 122
7.4. La stima delle <i>User Story</i> : Taglia T-Shirt e <i>Planning Poker</i>	» 124
7.4.1. Gli <i>Story Point</i> : definizione e importanza	» 128
7.5. L'identificazione e la stima dei <i>task</i>	» 129
7.6. L'aggiornamento dello <i>Sprint Backlog</i>	» 130

<b>8. I processi di implementazione</b>	pag.	132
8.1. L'implementazione dei <i>task</i> e lo Sprint Goal	»	132
8.2. L'utilizzo dello <i>Scrumboard</i>	»	134
8.3. Le metriche dello Sprint: la <i>Velocity</i> del team	»	135
8.4. <i>Burn Down</i> e <i>Burn Up Chart</i>	»	137
8.4.1. Esempi pratici di <i>Burn Down Chart</i> e la loro interpretazione	»	139
<b>9. I processi di Revisione, Retrospettiva e la chiusura del progetto Agile</b>	»	144
9.1. Dimostrare e validare lo Sprint	»	144
9.1.1. La dimostrazione e la Validazione dello Sprint: gli obiettivi	»	144
9.1.2. La fase preparatoria della Revisione	»	145
9.1.3. La fase esecutiva della Revisione dello Sprint	»	147
9.1.4. La dimostrazione degli Incrementi di prodotto	»	147
9.1.5. La raccolta del feedback degli stakeholder	»	149
9.1.6. I ruoli e le possibili criticità che possono emergere durante la Revisione	»	150
9.2. Condurre la Retrospettiva dello Sprint	»	151
9.2.1. L'attività preparatoria della Retrospettiva	»	152
9.2.2. La conduzione della Retrospettiva e le 5 azioni migliorative	»	154
9.2.3. I ruoli e le possibili criticità che possono emergere durante lo <i>Sprint Retrospective</i>	»	157
9.3. La chiusura del progetto Agile	»	160
9.3.1. La Retrospettiva del progetto: partecipanti e obiettivi	»	162
9.3.2. Il Report della Retrospettiva di progetto	»	164

# I principi del Project Management Agile



## 1.1. Project Management tradizionale e Agile: un parallelo

Nella gestione efficace dei progetti si riscontrano a livello nazionale e internazionale due tipologie di approcci: il primo, creato nel Novecento del secolo scorso è detto l'approccio "a cascata" (o *Waterfall*) mentre il successivo, detto Agile, vede la luce agli inizi degli anni 2000 e nasce per gestire progetti complessi in contesti di grande cambiamento.

La scelta della metodologia di gestione di un progetto è una decisione strategica che impatta profondamente sull'efficienza, sull'adattabilità e, in ultima analisi, sul successo di qualsiasi iniziativa. Sebbene l'approccio *Waterfall* e quello Agile siano entrambi volti al raggiungimento degli obiettivi progettuali, esistono differenze sostanziali negli approcci concettuali e nei processi di realizzazione.

Fondamentalmente, l'approccio più tradizionale ("a cascata") si basa su un'impostazione sequenziale e lineare: ad esempio nello sviluppo di un software le fasi sono la raccolta dei requisiti, la progettazione, l'implementazione, la fase di *testing*, il rilascio e l'installazione.

Proprio come l'acqua di una cascata non può risalire il suo corso ma fluisce naturalmente verso il basso, ogni fase del progetto *Waterfall* deve essere completata e approvata prima di passare alla successiva.

Questa rigidità comporta che i requisiti vengano fissati in modo esaustivo all'inizio del progetto, con un'enfasi marcata sulla documentazione dettagliata e formale in ogni passaggio.

Di conseguenza, la flessibilità al cambiamento è bassa e apportare modifiche significative, una volta avviate le fasi successive, risulta difficile e spesso costoso.

Il coinvolgimento del cliente è tipicamente limitato alle fasi iniziali di definizione dei requisiti e a quelle finali di accettazione. Il progresso è misu-

rato attraverso il completamento delle fasi e dei documenti e la consegna del prodotto avviene in un'unica soluzione alla fine del ciclo di vita del progetto.

L'approccio Agile invece adotta un'impostazione iterativa e incrementale, privilegiando la risposta rapida al cambiamento rispetto a una pianificazione rigida.

Le fasi del progetto sono sovrapposte e integrate in cicli brevi e ripetuti, noti come Sprint o iterazioni, che tipicamente durano da una a quattro settimane.

I requisiti non sono fissati a priori ma emergono e si evolvono continuamente attraverso un processo di apprendimento e adattamento: la flessibilità è un pilastro fondamentale dell'Agile, che accoglie e incorpora i cambiamenti anche in fasi avanzate del progetto, riconoscendo che il contesto del progetto può evolvere.

Il coinvolgimento del cliente è continuo e collaborativo attraverso la raccolta del suo feedback costante e tempestivo: il progresso è misurato dal software funzionante consegnato al termine di ogni iterazione e la consegna del prodotto è incrementale e frequente. L'obiettivo principale è la consegna continua di valore al cliente: i fallimenti sono visti come opportunità di apprendimento precoce, con costi di correzione relativamente contenuti.

Si riportano nella tabella sotto stante le principali differenze tra le due metodologie:

*Tab. 1 - Un parallelo tra gli approcci Waterfall e Agile*

<b>Caratteristica</b>	<b>Approccio Waterfall (a Cascata)</b>	<b>Approccio Agile</b>
Caratteristiche principali	Sequenziale, lineare, basato sulla pianificazione preventiva	Iterativo, incrementale, adattivo, basato sulla risposta al cambiamento
Fasi del progetto	Separate e distinte (requisiti, design, implementazione, test, consegna)	Sovrapposte e iterative (pianificazione, esecuzione, revisione, adattamento)
Requisiti degli stakeholder	Fissati all'inizio e documentati in modo esaustivo	Emergono e si evolvono durante il progetto
Flessibilità al cambiamento	Bassa; difficile e costoso gestire i cambiamenti una volta iniziata una fase	Alta; accetta e accoglie i cambiamenti anche in fasi avanzate
Coinvolgimento degli stakeholder	Limitato alle fasi iniziali (requisiti) e finali (accettazione)	Continuo e collaborativo durante tutto il ciclo di vita del progetto

Tab. 1 - segue

<b>Caratteristica</b>	<b>Approccio Waterfall (a Cascata)</b>	<b>Approccio Agile</b>
Consegna del prodotto	In un'unica soluzione alla fine del progetto	Incrementale, con rilasci frequenti e parziali del prodotto funzionante
Orientamento	Orientato al processo e alla documentazione	Orientato al valore e al prodotto funzionante
Team di progetto	Generalmente gerarchico, con ruoli e responsabilità rigidi	Auto-organizzato, multidisciplinare, con ruoli più fluidi
Tolleranza all'incertezza	Bassa: richiede alta certezza sui requisiti e sulla soluzione	Alta: prospera in contesti di incertezza e ambiguità
Misurazione del progresso	Basata sul completamento delle fasi e dei documenti	Basata sul software funzionante e sul feedback degli stakeholder
Gestione del rischio	Identificazione e mitigazione all'inizio del progetto	Apprendimento continuo dai rischi e adattamento costante
Documentazione	Essenziale, dettagliata e formale	Sufficiente, agile, spesso più concisa e orientata al valore
Test	Eseguiti principalmente nella fase finale dopo l'implementazione	Eseguiti in modo continuo e integrato in ogni iterazione
Feedback	Tardivo, principalmente al termine delle fasi principali	Continuo e tempestivo, in ogni iterazione
Pianificazione	Dettagliata e completa all'inizio	Adattiva e iterativa, con pianificazione di alto livello iniziale e dettagli emergenti
Frequenza delle revisioni	Punti di controllo formali alla fine di ogni fase (Milestones)	Revisioni frequenti (es. revisione dello Sprint) a intervalli regolari
Ruolo del Project Manager	Controllo rigido sul processo, budget, tempi e ambito	Non esiste un PM tradizionale, ma un Product Owner e uno Scrum Master
Obiettivo principale	Completare il progetto secondo il piano predefinito	Consegnare valore al cliente in modo continuo e sostenibile
Visione del fallimento	Il fallimento si manifesta spesso alla fine del progetto, essendo molto costoso	I fallimenti sono considerati opportunità di apprendimento precoce, con costi contenuti

### 1.1.1. L'analisi dei contesti di sviluppo di un progetto: il Modello *Cynefin*

Se entrambi i modelli mirano a concludere il progetto in modo efficace ed efficiente, quali sono dunque i parametri per scegliere l'uno o l'altro approccio?

Si ritiene che l'elemento determinante per la scelta sia il contesto entro il quale il progetto si colloca: il Modello *Cynefin* (termine gallese che significa “luogo” o “habitat”), creato da Dave Snowden<sup>1</sup>, individua 5 diversi contesti di progetti (chiamati Domini):

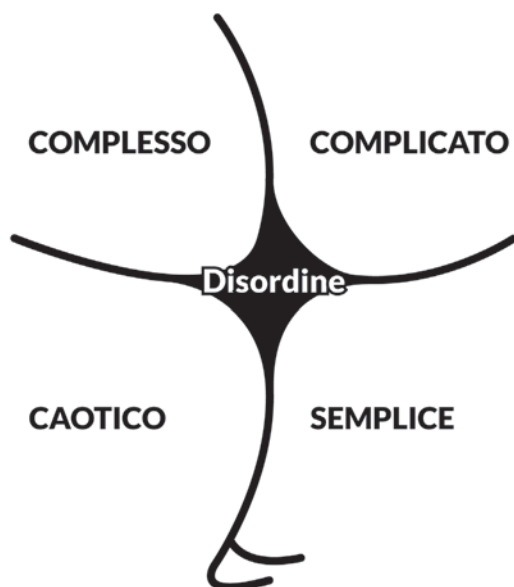


Fig. 1 - Il Modello *Cynefin*

Nel dominio semplice (altrimenti chiamato “chiaro, ovvio”), le relazioni causa-effetto sono chiare, prevedibili, ripetibili e riconosciute da tutti: le risposte sono standardizzate e le *best practice* possono essere applicate. È una situazione in cui le situazioni sono semplici da gestire e le soluzioni sono dirette.

1. Per un approfondimento si veda Snowden Dave, “Cynefin, weaving sense, making into the fabric of our world”, Cognitive Edge, 2020.

Tab. 2 - Esempi di progetti di dominio semplice

<b>Progetto</b>	<b>Caratteristiche</b>
Sviluppo di un sito web aziendale con requisiti chiari e definiti	Ogni fase, può essere completata in sequenza, facilitando il controllo del progetto e la garanzia di rispettare le scadenze
Costruzione di infrastrutture (es. edifici e ponti)	Ogni fase della progettazione deve seguire un approccio sequenziale rigoroso, con tutte le valutazioni e i permessi necessari completati prima di passare alla fase successiva
Organizzazione di un evento aziendale interno con <i>format</i> consolidato	Ad esempio, la festa di Natale annuale che segue un copione simile ogni anno
Sostituzione di un hardware di rete obsoleto con un modello equivalente	Il processo di migrazione e configurazione è ben documentato dal fornitore
<i>Onboarding</i> di un nuovo dipendente con un ruolo standardizzato	Si segue una <i>check list</i> consolidata di passaggi (contratto, accesso IT, presentazione team, ecc.)
Progetti di ricerca e sviluppo	I progetti di sviluppo di un nuovo prodotto chimico o farmaceutico prevedono delle fasi sequenziali che devono essere seguite rigorosamente per garantire il rispetto delle norme e la sicurezza

In questo dominio, gli esperti concordano che il modello più efficace sia il *Waterfall*: la natura intrinseca del contesto semplice, caratterizzato da requisiti stabili e ben compresi, risultati prevedibili e la presenza di “migliori pratiche” consolidate, si allinea perfettamente con la metodologia “a cascata”.

Il processo sequenziale e la forte enfasi sulla pianificazione iniziale e sulla documentazione dettagliata offrono la massima efficienza, controllo e prevedibilità in un ambiente dove le sorprese sono minime.

Il secondo dominio è quello complicato ed è caratterizzato da relazioni causa-effetto non immediatamente evidenti che richiedono analisi e conoscenze specialistiche per essere individuate.

È una situazione in cui non esistono “*best practice*” ma “*good practice*”, ovvero sono possibili diverse soluzioni.

Tab. 3 - Esempi di progetti di dominio complicato

<b>Progetto</b>	<b>Caratteristiche</b>
Sviluppo di software per settori regolamentati	Progetti di software per ambiti come la finanza o la sanità, dove i requisiti normativi e le specifiche tecniche devono essere scrupolosamente rispettati
Progetti di integrazione di sistemi IT	Quando un'organizzazione deve integrare diversi sistemi esistenti in modo che funzionino insieme (ad esempio, sistemi di gestione delle relazioni con i clienti e software di contabilità)
Sviluppo di prodotti meccanici complessi	La progettazione di dispositivi meccanici o elettronici con requisiti stringenti, come apparecchiature mediche
Sviluppo di applicazioni web innovative	Progetti di sviluppo di nuove applicazioni web in mercati in rapida evoluzione, dove i requisiti degli utenti possono cambiare frequentemente
Progetti di marketing digitale	Campagne di marketing che richiedono continui aggiustamenti e ottimizzazioni in base all'analisi delle performance (come SEO, contenuti e pubblicità)
Innovazione nel design di prodotti	Sviluppo di nuovi prodotti o servizi dove l'innovazione è fondamentale e dove la prototipazione rapida e i feedback da parte degli utenti finali possono informare le successive iterazioni del <i>design</i>

Nei progetti che rientrano nel dominio complicato, si consiglia un *mix* ponderato dei due approcci: nella tabella sopra riportata i primi 3 esempi suggeriscono un ricorso al metodo “a cascata”, mentre gli ultimi 3 possono essere gestiti in modo più efficace grazie all'approccio Agile.

In un dominio complesso, le relazioni causa-effetto non sono immediatamente evidenti e si possono manifestare solo a posteriori. Le soluzioni emergono attraverso l'esperimento e l'adattamento: non esistono infatti soluzioni predefinite e l'ambiente è dinamico e in continua evoluzione.

Tab. 4 - Esempi di progetti di dominio complesso

<b>Progetto</b>	<b>Caratteristiche</b>
Lancio di una start up in un mercato innovativo e non ancora definito	La soluzione emerge attraverso esperimenti continui e cambi di direzione improvvisi
Sviluppo di un nuovo prodotto o servizio di rottura ( <i>disruptive</i> ) altamente innovativo	I requisiti e la domanda di mercato evolvono man mano che il prodotto viene testato
Sviluppo di una strategia di sostenibilità aziendale	Non c'è un percorso unico, richiede iterazioni, differenti strategie di <i>stakeholder engagement</i> e risposte a eventuali nuove normative
Gestione di una crisi di reputazione aziendale in un'era di social media	La risposta deve essere dinamica, basata sul monitoraggio delle reazioni e sull'adattamento del messaggio in tempo reale
Definizione della visione a lungo termine di un'azienda in un settore in rapida evoluzione	Non si può pianificare in anticipo, si deve esplorare il futuro attraverso scenari e prototipi
Entrata in un nuovo mercato geografico con alta incertezza regolamentare	Richiede piccoli esperimenti di marketing, adattamento dei prodotti e apprendimento continuo dal feedback dei clienti

La natura emergente e imprevedibile di un contesto complesso rende l'approccio *Waterfall* non solo inefficace ma potenzialmente dannoso a causa della definizione rigida dei requisiti e la produzione di una documentazione corposa che deve essere continuamente rivista e aggiornata.

Gli esperti concordano che in tale dominio l'approccio più indicato è quello Agile grazie alla sua enfasi sulla sperimentazione, il feedback rapido e l'adattamento continuo.

In un dominio caotico non c'è alcuna relazione causa-effetto evidente, i problemi cambiano rapidamente e la situazione è altamente instabile: l'obiettivo principale è agire velocemente per ripristinare l'ordine e stabilizzare la situazione

Esempi di tale dominio sono le crisi sanitarie ed epidemie, le minacce alla sicurezza informatica aziendale, i disastri naturali, le situazioni di conflitto e di forte instabilità politica, l'interruzione totale della catena di fornitura a livello globale, la fuga di dati sensibili della clientela che diventa di dominio pubblico, il blocco completo della produzione a causa di un guasto imprevisto e non diagnosticabile.

In questi casi l'approccio *Waterfall* e *Agile* sono sconsigliati dagli esperti che propendono invece per un approccio *ad hoc* di tipo “comando e controllo”, rapido e decisionista.

Una volta che la situazione è stata stabilizzata e un minimo di ordine è stato ristabilito (cioè, quando il contesto si sposta dal dominio caotico a quello complesso o complicato), solo allora si potrà considerare l'introduzione di metodologie più strutturate come l'*Agile*, per gestire l'emergere di soluzioni o per affrontare la complessità residua.

L'ultimo dominio del modello *Cynefin* è detto disordinato e riguarda situazioni in cui non è chiaro a quale degli altri domini la situazione appartenga.

Non ci sono relazioni causa-effetto evidenti e la situazione è caratterizzata da forte incertezza e ambiguità.

È un'area di confusione e richiede un'analisi per classificare il problema e scegliere la risposta appropriata: la priorità è quella di scomporre la situazione per poterla ricondurre a uno degli altri domini.

Tab. 5 - Esempi di dominio disordinato

Situazioni	Possibili mix
Alta rotazione del personale in un reparto specifico senza una causa apparente	Non si capisce se sia un problema di leadership (complesso) o di processi (complicato)
Risultati di vendita in calo significativo, ma non si riesce a identificare la causa principale	Potrebbe essere il mercato (complesso), il prodotto (complicato) o la motivazione della forza vendita (semplice/complicato)
Il lancio di un nuovo prodotto sul mercato ha risultati altalenanti e inspiegabili	Non si sa se il problema è nel prodotto (complicato), nel marketing (complesso) o nella domanda (caotico/complesso)
Un'acquisizione aziendale sta generando più problemi di integrazione di quanto previsto, ma non si capisce dove sia il blocco principale	Potrebbe essere una difficoltà ad armonizzare le culture aziendali (complesso), l'integrazione dei diversi sistemi (complicato) o resistenze individuali (semplice)
Il morale di un team di progetto è basso e la produttività è in declino, ma nessuno riesce a definire il motivo:	Potrebbe essere lo stile di leadership del Project Manager (complesso), il carico di lavoro mal distribuito (semplice/complicato) o mancanza di strumenti operativi (complicato)

Il dominio disordinato è pericoloso perché può portare a decisioni incoerenti o basate su assunzioni errate: l'obiettivo primario aziendale è quello di “scomporre” la situazione in problemi più piccoli o di effettuare azioni mirate a capire la natura del contesto.

Una volta compresa meglio la natura dei diversi problemi questi si possono classificare in uno degli altri quattro domini (Semplice, Complicato, Complesso o Caotico) e solo allora si potrà scegliere la metodologia appropriata: per i problemi semplici si potrebbe optare per *Waterfall*, se invece sono complicati o complessi è preferibile l'Agile.

Se dovesse persistere il caos, allora si suggerisce l'approccio “comando e controllo” per stabilizzare la situazione.

### 1.1.2. Test di auto-valutazione: “La mia azienda è pronta per il Project Management Agile?”

Nel caso si debbano affrontare progetti in contesti complicati e complessi si è visto precedentemente come il *Framework* Agile sia quello più adatto.

Per capire se la vostra azienda è pronta ad implementare tale approccio nella gestione dei progetti oppure è preferibile una formula mista (ibrida), si riporta di seguito un questionario di auto-valutazione (*Self Assessment*) proposto dal Project Management Institute (PMI)<sup>2</sup>.

Tab. 6 - Test di auto-valutazione

Domande	Scheda di valutazione	Valutazione (scala 1-10)
Esiste in azienda uno sponsor con adeguata seniority che comprende e sostiene l'utilizzo di un approccio Agile per il progetto	1 = Sì 5 = Forse 10 = No	
Gli sponsor e i manager che lavoreranno con il team garantiscono supporto e feedback continui durante il progetto?	1 = Sì 5 = Probabile 10 = Improbabile	

2. “Guida alla pratiche dell'Agile”, PMI e Agile Alliance, 2017.