

PERCORSO FORMATIVO — PRODUCT MANAGER

**Dispensa 7**

## **La Collaborazione con il Team Tecnico e il Debito Tecnico**

*Costruire fiducia con sviluppatori e designer, e comprendere perché anche la manutenzione merita spazio nella roadmap.*

---

**A cura di Marco Consiglio**

Formatore e Course Director — Contenuti aggiornati al 2026

## Indice

---

|                                                                                  |    |
|----------------------------------------------------------------------------------|----|
| 1. Introduzione e obiettivi didattici .....                                      | 3  |
| 2. Costruire fiducia senza autorità gerarchica.....                              | 4  |
| 3. Comprendere, senza dover programmare, la logica dello sviluppo software ..... | 6  |
| 4. Scrivere requisiti chiari: dalla user story ai criteri di accettazione .....  | 9  |
| 5. Il debito tecnico: che cos'è e perché riguarda anche il Product Manager.....  | 11 |
| 6. Negoziare spazio per la manutenzione nella roadmap .....                      | 13 |
| 7. Collaborare efficacemente con il design .....                                 | 15 |
| 8. Un principio di rispetto reciproco tra competenze diverse .....               | 17 |
| 9. Caso pratico guidato.....                                                     | 18 |
| 10. Glossario dei termini chiave.....                                            | 20 |
| 11. Checklist finale e autoverifica.....                                         | 21 |
| 12. Riferimenti e risorse aggiornate al 2026 .....                               | 22 |

## 1. Introduzione e obiettivi didattici

---

Questa dispensa prosegue il percorso Product Manager, approfondendo un tema già più volte accennato nelle dispense precedenti: la collaborazione quotidiana con il team tecnico, già presentata nella Dispensa 1 di questo percorso come una delle sfide più significative del ruolo, proprio per l'assenza di autorità gerarchica diretta su questa componente essenziale del lavoro di prodotto.

Applicando quanto già presentato nella Dispensa 1 di questo percorso a proposito del quarto luogo comune sfatato, questa dispensa mostra concretamente come un Product Manager privo di un background tecnico specifico possa comunque costruire un dialogo efficace con sviluppatori e designer, e approfondisce un tema tecnico specifico di grande rilevanza pratica: il debito tecnico, e la necessità di negoziare per esso uno spazio adeguato nella roadmap già presentata nella Dispensa 5 di questo percorso.

### A chi si rivolge questa dispensa

Si rivolge a chi ha seguito le prime sei dispense di questo percorso e desidera acquisire, partendo da zero, gli strumenti pratici per collaborare efficacemente con il team tecnico, senza necessariamente possedere competenze di programmazione.

### Obiettivi di apprendimento

1. Comprendere come costruire fiducia con il team tecnico senza autorità gerarchica;
2. Acquisire una comprensione di base della logica dello sviluppo software, senza dover programmare;
3. Saper scrivere requisiti chiari attraverso user story e criteri di accettazione;
4. Comprendere che cos'è il debito tecnico e perché riguarda anche il Product Manager;
5. Saper negoziare spazio per la manutenzione nella roadmap;
6. Saper collaborare efficacemente con il design, e comprendere il principio del rispetto reciproco tra competenze diverse.

## 2. Costruire fiducia senza autorità gerarchica

Applicando quanto già presentato nella Dispensa 1 di questo percorso a proposito del principio di guidare senza comandare, approfondiamo ora concretamente come un Product Manager costruisce, nella pratica quotidiana, una relazione di fiducia con il team tecnico, l'elemento fondamentale su cui si basa l'intera efficacia della collaborazione presentata in questa dispensa.

### Il rispetto per la competenza altrui come punto di partenza

*Applicando quanto già presentato nella Dispensa 1 di questo percorso a proposito del triangolo di responsabilità, un Product Manager efficace riconosce esplicitamente, nel proprio comportamento quotidiano, che il team tecnico possiede una competenza specialistica che il Product Manager stesso non possiede allo stesso livello: questo riconoscimento, tutt'altro che scontato, si traduce concretamente nel chiedere piuttosto che imporre soluzioni tecniche specifiche, e nel fidarsi del giudizio tecnico del team sulla fattibilità e sulla migliore implementazione di una funzionalità richiesta.*

### La trasparenza sulle motivazioni delle richieste

Applicando quanto già presentato nella Dispensa 2 di questo percorso a proposito della comunicazione della visione, un Product Manager costruisce fiducia comunicando sempre il perché di una richiesta, non solo il cosa: un team che comprende la motivazione sottostante a una funzionalità richiesta è più propenso a proporre soluzioni tecniche alternative, talvolta migliori di quella originariamente immaginata dal Product Manager stesso, un beneficio concreto della trasparenza che va oltre la sola costruzione della relazione.

### La coerenza nel tempo come fondamento della credibilità

Applicando quanto già presentato nella Dispensa 4 di questo percorso a proposito della coerenza nel processo di prioritizzazione, un Product Manager che cambia frequentemente idea, senza una motivazione chiara, o che promette al team una stabilità nelle priorità per poi disattenderla sistematicamente, erode progressivamente la fiducia necessaria per una collaborazione efficace, un rischio che rende ancora più rilevante quanto già presentato nella Dispensa 5 di questo percorso a proposito della roadmap orientata ai temi, meno soggetta a stravolgimenti improvvisi rispetto a un approccio basato su date rigide.

### Il valore della presenza costante, non solo nei momenti di decisione

Applicando quanto già presentato nella Dispensa 6 del percorso Business Manager di questa collana a proposito della leadership di prossimità, un Product Manager che partecipa con regolarità alle attività quotidiane del team (le riunioni di sincronizzazione dello sprint già presentato nel capitolo 3 di questa dispensa, i momenti informali di confronto tecnico) costruisce una relazione di fiducia più solida rispetto a un Product Manager che si presenta solo nei momenti di decisione formale, come la definizione delle priorità o la revisione dei risultati.

Questa presenza costante permette anche di cogliere tempestivamente segnali di difficoltà o frustrazione all'interno del team, prima che si trasformino in problemi più seri, secondo lo stesso principio di ascolto attivo già presentato nella Dispensa 3 di questo percorso applicato in questo caso al proprio team interno, non solo agli utenti esterni del prodotto.

### 3. Comprendere, senza dover programmare, la logica dello sviluppo software

---

Applicando quanto già presentato nella Dispensa 1 di questo percorso a proposito del primo luogo comune sfatato, presentiamo ora alcuni concetti tecnici essenziali che un Product Manager dovrebbe comprendere per dialogare efficacemente con il team, senza per questo dover acquisire competenze di programmazione vera e propria.

#### Il concetto di sprint e di iterazione

Molte organizzazioni di sviluppo software lavorano secondo cicli di lavoro brevi e ricorrenti, tipicamente di una o due settimane, chiamati sprint, al termine dei quali il team rilascia un incremento di lavoro completato e verificabile: comprendere questo ritmo aiuta un Product Manager a calibrare le proprie aspettative e a pianificare la comunicazione della roadmap già presentata nella Dispensa 5 di questo percorso in modo coerente con il reale ciclo di lavoro del team.

#### La differenza tra frontend e backend

Un concetto tecnico di base, utile per comprendere le stime di sforzo fornite dal team: il frontend è la parte del software con cui l'utente interagisce direttamente (l'interfaccia visiva), mentre il backend è l'infrastruttura non visibile che elabora i dati e la logica sottostante. Una funzionalità apparentemente semplice dal punto di vista dell'interfaccia utente può richiedere un lavoro backend significativamente più complesso, un elemento che spiega perché le stime di sforzo talvolta sorprendono un Product Manager privo di questa comprensione tecnica di base.

#### Il concetto di rilascio incrementale (MVP)

*Applicando quanto già presentato nella Dispensa 4 di questo percorso a proposito del framework RICE e del parametro effort, un principio tecnico centrale nello sviluppo software moderno riguarda la possibilità di rilasciare una versione minima ma funzionante di una funzionalità (Minimum Viable Product, o più specificamente per una singola funzionalità, un rilascio incrementale), per poi arricchirla progressivamente sulla base del riscontro reale degli utenti, secondo gli stessi principi di ricerca continuativa già presentati nella Dispensa 3 di questo percorso. Un Product Manager che comprende questo principio può negoziare con il team rilasci più rapidi di versioni semplificate, invece di attendere una versione completa e più complessa da realizzare, riducendo il tempo necessario per verificare concretamente se una direzione di sviluppo sia realmente quella giusta.*

#### Un principio pratico: fare domande, non fingere di sapere

Applicando quanto già presentato nella Dispensa 1 di questo percorso a proposito della resilienza di fronte all'incertezza, un Product Manager privo di un background tecnico specifico non dovrebbe mai fingere di comprendere pienamente un concetto tecnico che in realtà non

padroneggia: fare domande dirette e genuine al team tecnico, anche quando potrebbero apparire elementari, costruisce nel tempo sia una maggiore comprensione tecnica personale, sia una relazione di fiducia autentica, ben più solida di quella costruita fingendo una competenza non realmente posseduta.

### **Un ulteriore concetto utile: l'ambiente di test e l'ambiente di produzione**

Un ultimo concetto tecnico di base, utile per comprendere il linguaggio quotidiano del team, riguarda la distinzione tra ambiente di test (o di sviluppo) e ambiente di produzione: il primo è uno spazio isolato in cui il team verifica il funzionamento di una nuova funzionalità prima che sia effettivamente disponibile agli utenti reali, il secondo è la versione del prodotto realmente utilizzata dagli utenti finali. Comprendere questa distinzione aiuta un Product Manager a interpretare correttamente affermazioni frequenti nel dialogo quotidiano con il team, come funziona in test ma non ancora in produzione, evitando fraintendimenti su quanto una funzionalità sia realmente pronta per essere comunicata agli utenti secondo gli strumenti di roadmap già presentati nella Dispensa 5 di questo percorso.

## 4. Scrivere requisiti chiari: dalla user story ai criteri di accettazione

Presentiamo ora uno strumento tecnico specifico e di grande rilevanza pratica quotidiana: la user story, il formato attraverso cui un Product Manager comunica al team tecnico che cosa sviluppare, integrando gli elementi già presentati nella Dispensa 3 di questo percorso a proposito della comprensione dei bisogni utente.

### La struttura tipica di una user story

#### Struttura tipica di una user story

Come [tipo di utente], voglio [azione desiderata], così da [beneficio atteso].

### Un esempio applicato al caso già presentato in questo percorso

#### Esempio di user story

Come piccolo produttore agricolo, voglio vedere immediatamente quante unità di un prodotto ho ancora disponibili, così da non promettere ai clienti quantità che non posso effettivamente consegnare.

### Il valore di questo formato

*Applicando quanto già presentato nella Dispensa 3 di questo percorso a proposito della distinzione tra richieste esplicite e bisogni reali, il formato della user story costringe esplicitamente chi la scrive a specificare non solo la funzionalità desiderata, ma anche chi ne beneficia e perché, un promemoria costante del bisogno reale sottostante che previene il rischio, già presentato in quella dispensa, di costruire funzionalità basate su richieste superficiali prive di un reale fondamento nei bisogni degli utenti.*

### I criteri di accettazione: definire quando una funzionalità è completa

Accanto alla user story, i criteri di accettazione specificano condizioni concrete e verificabili che devono essere soddisfatte perché la funzionalità sia considerata effettivamente completa: applicando quanto già presentato nella Dispensa 4 di questo percorso a proposito del modello Kano, questi criteri dovrebbero riflettere esplicitamente gli elementi che generano soddisfazione per l'utente, non solo la sola presenza tecnica della funzionalità.

### Un esempio di criteri di accettazione

- Il produttore vede il livello di scorte aggiornato in tempo reale sulla schermata principale;
- Se le scorte scendono sotto una soglia configurabile, viene mostrato un segnale visivo di avviso;
- Il produttore può aggiornare manualmente le scorte anche in assenza di connessione internet, con una sincronizzazione automatica al ripristino della connessione.

**Un principio pratico: coinvolgere il team nella scrittura, non solo nella lettura**

Applicando quanto già presentato nel capitolo 2 di questa dispensa a proposito del rispetto per la competenza tecnica, la scrittura di user story e criteri di accettazione non dovrebbe essere un'attività isolata del solo Product Manager, comunicata al team come un documento già completo e non negoziabile: coinvolgere gli sviluppatori nella revisione e nel perfezionamento dei criteri di accettazione, prima dell'avvio effettivo dello sviluppo, permette di identificare precocemente ambiguità o casi limite non ancora considerati, riducendo significativamente il rischio di fraintendimenti che emergerebbero altrimenti solo a sviluppo già avanzato, quando le correzioni risultano tipicamente più costose e complesse da implementare.

## 5. Il debito tecnico: che cos'è e perché riguarda anche il Product Manager

Presentiamo ora un concetto tecnico centrale, spesso poco compreso da chi si avvicina per la prima volta al ruolo di Product Manager, ma di grande rilevanza pratica per la costruzione di una roadmap sostenibile già presentata nella Dispensa 5 di questo percorso: il debito tecnico.

### Una definizione operativa

Il debito tecnico è l'insieme delle scelte tecniche non ottimali, spesso adottate consapevolmente per accelerare un rilascio nel breve periodo, che nel tempo rallentano progressivamente lo sviluppo futuro, richiedendo un investimento di manutenzione e correzione crescente se non gestito con attenzione: una metafora finanziaria efficace, che richiama quanto già presentato nella Dispensa 5 del percorso Finanza in Azienda di questa collana a proposito degli strumenti di finanziamento, poiché come un debito finanziario, il debito tecnico comporta un costo implicito (interessi) che si accumula nel tempo se non affrontato.

### Perché il debito tecnico riguarda direttamente il Product Manager

*Applicando quanto già presentato nella Dispensa 4 di questo percorso a proposito della prioritizzazione, un errore frequente consiste nel considerare il debito tecnico esclusivamente un problema del team di sviluppo, da affrontare autonomamente senza coinvolgere il Product Manager: nella realtà, poiché il debito tecnico compete direttamente per le stesse risorse di sviluppo necessarie per nuove funzionalità, la decisione su quanto tempo dedicare alla sua gestione è, a tutti gli effetti, una decisione di prioritizzazione che il Product Manager dovrebbe presidiare attivamente, non delegare completamente al solo team tecnico.*

### Un esempio dell'impatto del debito tecnico non gestito

| Situazione                                       | Conseguenza tipica                                 |
|--------------------------------------------------|----------------------------------------------------|
| Debito tecnico ignorato per lungo tempo          | Velocità di sviluppo che rallenta progressivamente |
| Debito tecnico gestito con investimenti regolari | Velocità di sviluppo mantenuta stabile nel tempo   |

## 6. Negoziare spazio per la manutenzione nella roadmap

Applicando quanto già presentato nel capitolo 5 di questa dispensa a proposito del debito tecnico, e quanto già presentato nella Dispensa 5 di questo percorso a proposito della costruzione della roadmap, approfondiamo ora concretamente come un Product Manager negozia esplicitamente uno spazio per la manutenzione e la gestione del debito tecnico all'interno della propria roadmap.

### Un principio pratico: allocare una quota fissa di capacità

Applicando quanto già presentato nella Dispensa 5 di questo percorso a proposito del margine esplicito per l'incertezza, una pratica diffusa consiste nell'allocare consapevolmente, all'interno della capacità di sviluppo complessiva disponibile in ciascun ciclo, una quota fissa (tipicamente tra il dieci e il venti per cento, variabile in funzione della specifica situazione tecnica dell'organizzazione) dedicata esplicitamente alla manutenzione e alla gestione del debito tecnico, piuttosto che considerarla un'attività residuale da svolgere solo se avanza tempo dopo lo sviluppo di nuove funzionalità.

### Comunicare il valore della manutenzione agli stakeholder non tecnici

*Applicando quanto già presentato nella Dispensa 6 di questo percorso a proposito della comunicazione a pubblici diversi, un Product Manager dovrebbe saper comunicare efficacemente, anche a interlocutori non tecnici come la direzione aziendale, perché investire tempo nella manutenzione, apparentemente priva di un beneficio immediato e visibile per l'utente, rappresenti in realtà un investimento necessario per la sostenibilità futura del ritmo di sviluppo, secondo lo stesso principio già presentato nella Dispensa 7 di questo percorso di Finanza in Azienda di questa collana a proposito del capital budgeting, in cui alcuni investimenti generano un ritorno non immediato ma essenziale per la salute complessiva dell'organizzazione nel medio periodo.*

### Un principio di equilibrio, non di eccesso opposto

Applicando lo stesso principio di proporzionalità già più volte richiamato in questo percorso, un Product Manager dovrebbe evitare sia l'errore di ignorare completamente il debito tecnico, sia l'errore opposto di lasciare che la sola manutenzione tecnica assorba una quota eccessiva delle risorse disponibili, paralizzando lo sviluppo di nuove funzionalità realmente necessarie per gli utenti: un dialogo costante e trasparente con il team tecnico, secondo quanto già presentato nel capitolo 2 di questa dispensa, aiuta a calibrare correttamente questo equilibrio nel tempo.

### Un principio di prioritizzazione anche all'interno del debito tecnico

Applicando quanto già presentato nella Dispensa 4 di questo percorso a proposito degli strumenti di prioritizzazione, non tutto il debito tecnico accumulato ha lo stesso livello di urgenza o di impatto: applicando una logica concettualmente simile al framework RICE già presentato in

quella dispensa, un Product Manager dovrebbe collaborare con il team tecnico per identificare quali elementi di debito tecnico stiano generando il maggiore impatto negativo sulla velocità di sviluppo o sulla stabilità del prodotto, concentrando le risorse limitate allocate alla manutenzione su questi elementi prioritari, piuttosto che affrontare il debito tecnico in modo indiscriminato senza una logica di priorizzazione altrettanto rigorosa di quella applicata alle nuove funzionalità.

## 7. Collaborare efficacemente con il design

Applicando quanto già presentato nella Dispensa 1 di questo percorso a proposito del triangolo di responsabilità, concludiamo la panoramica sulla collaborazione tecnica con un approfondimento specifico sul dialogo con la funzione design, distinta dal solo team di sviluppo già presentato nei capitoli precedenti di questa dispensa.

### Il ruolo specifico del design nel triangolo di responsabilità

Applicando quanto già presentato nella Dispensa 1 di questo percorso, la funzione design presidia specificamente la componente dell'esperienza utente all'interno del triangolo di responsabilità, con competenze specifiche (usabilità, coerenza visiva, accessibilità) che integrano, senza sovrapporsi, la comprensione dei bisogni utente che il Product Manager costruisce attraverso la ricerca già presentata nella Dispensa 3 di questo percorso.

### Un principio di collaborazione precoce, non sequenziale

*Un errore frequente, specie in organizzazioni meno mature nella collaborazione tra funzioni diverse, consiste nel trattare il design come una fase sequenziale successiva alla decisione del Product Manager (prima si decide che cosa costruire, poi il design definisce come dovrebbe apparire): un approccio più efficace, secondo gli stessi principi di collaborazione già presentati nel capitolo 2 di questa dispensa, coinvolge il design fin dalle fasi iniziali di esplorazione di un problema, poiché la sensibilità specifica del designer verso l'esperienza utente può arricchire significativamente la comprensione stessa del problema da risolvere, non solo la sua soluzione visiva finale.*

### Un principio di rispetto per la competenza specifica del design

Applicando lo stesso principio già presentato nel capitolo 2 di questa dispensa a proposito del rispetto per la competenza tecnica del team di sviluppo, un Product Manager dovrebbe evitare di imporre soluzioni di design specifiche basate sulle proprie preferenze personali, privilegiando invece un dialogo che valorizzi la competenza specialistica del designer, secondo lo stesso principio di guida senza comando già presentato nella Dispensa 1 di questo percorso come elemento distintivo dell'intero ruolo.

### Gestire il feedback sul design con la stessa disciplina della ricerca utente

Un ultimo elemento pratico riguarda come un Product Manager gestisce il proprio feedback su una proposta di design ricevuta dal team: applicando quanto già presentato nella Dispensa 3 di questo percorso a proposito della distinzione tra opinioni personali e bisogni reali degli utenti, un feedback basato sulla sola preferenza estetica personale del Product Manager (non mi piace questo colore) ha un valore molto inferiore rispetto a un feedback ancorato esplicitamente ai bisogni degli utenti già identificati attraverso la ricerca (i nostri utenti, secondo le interviste condotte, faticano a distinguere visivamente lo stato critico delle scorte). Un Product Manager

dovrebbe abituarsi a formulare il proprio feedback sempre in questi termini, riconducendolo esplicitamente alla ricerca utente già presentata in questo percorso, piuttosto che al proprio gusto personale, che non è necessariamente rappresentativo del pubblico target del prodotto.

## 8. Un principio di rispetto reciproco tra competenze diverse

Concludiamo la parte teorica di questa dispensa con una riflessione di sintesi, che collega tutti gli elementi già presentati nei capitoli precedenti attorno a un principio comune: il rispetto reciproco tra competenze professionali diverse, ciascuna insostituibile nel proprio ambito specifico.

### Il Product Manager come integratore, non come esperto in ogni ambito

*Applicando quanto già presentato nella Dispensa 1 di questo percorso a proposito del triangolo di responsabilità, un Product Manager efficace non aspira a diventare un esperto di sviluppo software o di design al pari degli specialisti con cui collabora quotidianamente, un obiettivo tanto irrealistico quanto controproducente, ma sviluppa una comprensione sufficiente di ciascun ambito per dialogare efficacemente, integrando competenze diverse verso un obiettivo condiviso, secondo il ruolo di sintesi già presentato in quella dispensa come elemento centrale e distintivo del proprio lavoro.*

### Un principio di sintesi per l'intera dispensa

#### Un principio di sintesi per l'intera dispensa

La collaborazione con il team tecnico, in tutte le dimensioni presentate in questa dispensa — la costruzione di fiducia senza autorità gerarchica, la comprensione di base della logica dello sviluppo, gli strumenti per scrivere requisiti chiari, la gestione del debito tecnico, la negoziazione dello spazio per la manutenzione, la collaborazione con il design, il rispetto reciproco tra competenze diverse — rappresenta uno degli ambiti in cui il giudizio maturato e le competenze relazionali già presentate nella Dispensa 1 di questo percorso si traducono più concretamente nella pratica quotidiana, spesso più decisive per il successo del ruolo rispetto alla sola competenza tecnica specifica.

### Un ultimo elemento: la collaborazione come investimento che si ripaga nel tempo

Un'ultima riflessione conclude questa dispensa: la costruzione di una relazione di fiducia solida con il team tecnico e con il design, secondo tutti gli elementi già presentati in questa dispensa, richiede un investimento di tempo ed energia che può apparire, specialmente nelle prime settimane di collaborazione con un nuovo team, meno urgente rispetto alle pressioni quotidiane di prioritizzazione e consegna già presentate nelle dispense precedenti di questo percorso. Applicando lo stesso principio di investimento nel lungo periodo già presentato nella Dispensa 6 di questo percorso a proposito della manutenzione tecnica, questo tempo dedicato alla relazione non è mai tempo sottratto al lavoro produttivo, ma un investimento che si ripaga ampiamente nel tempo, sotto forma di una collaborazione più fluida, di soluzioni tecniche più creative, e di una resilienza organizzativa superiore nei momenti in cui, inevitabilmente, qualcosa non procede secondo i piani originari.

## 9. Caso pratico guidato

---

Concludiamo la parte teorica con un caso pratico semplificato, che applica gli strumenti presentati in questa dispensa.

### Il contesto

Riprendiamo il caso già presentato nelle dispense precedenti di questo percorso: la Product Manager della piattaforma per agricoltori, dopo i primi mesi di sviluppo secondo la roadmap già presentata nella Dispensa 5 di questo percorso, riceve dal team tecnico una segnalazione esplicita di crescente debito tecnico accumulato durante la corsa iniziale al primo rilascio, che sta iniziando a rallentare lo sviluppo di nuove funzionalità.

### Passo 1 — Ascolto e comprensione della segnalazione tecnica

Applicando gli strumenti del capitolo 2, la Product Manager, pur non possedendo un background tecnico specifico secondo quanto già presentato nella Dispensa 1 di questo percorso, dedica tempo ad ascoltare con attenzione le spiegazioni del team, facendo domande dirette per comprendere realmente l'impatto pratico del problema segnalato, secondo il principio già presentato nel capitolo 3 di questa dispensa.

### Passo 2 — Scrittura di user story chiare per le prossime funzionalità

Applicando gli strumenti del capitolo 4, mentre negozia lo spazio per la manutenzione, la Product Manager continua a preparare user story ben strutturate per le funzionalità successive già presenti nella roadmap, con criteri di accettazione chiari che riducono il rischio di ambiguità nella fase di sviluppo.

### Passo 3 — Negoziazione di uno spazio esplicito per il debito tecnico

Applicando gli strumenti dei capitoli 5 e 6, viene negoziata con il team un'allocazione del quindici per cento della capacità di sviluppo dei prossimi due trimestri dedicata esplicitamente alla riduzione del debito tecnico accumulato, un impegno che la Product Manager comunica con trasparenza anche alla direzione aziendale, spiegando le ragioni di questo investimento apparentemente privo di beneficio immediato per l'utente.

### Passo 4 — Coinvolgimento tempestivo del design nelle funzionalità successive

Applicando gli strumenti del capitolo 7, per le funzionalità pianificate nel trimestre successivo, la Product Manager coinvolge il designer fin dalla fase di esplorazione del problema, non solo nella fase di definizione visiva successiva, ottenendo un contributo prezioso sulla reale esperienza d'uso della funzionalità di visualizzazione delle scorte già presentata nelle dispense precedenti di questo percorso.

## Passo 5 — Un miglioramento misurabile nel tempo

Applicando gli strumenti della Dispensa 6 di questo percorso, dopo i due trimestri di investimento nella riduzione del debito tecnico, il team riporta una velocità di sviluppo significativamente migliorata, un risultato che la Product Manager comunica alla direzione come conferma della correttezza della decisione presa, rafforzando ulteriormente la fiducia reciproca con il team tecnico già costruita nel corso di questo processo.

### Che cosa insegna questo caso

Il caso mostra come una Product Manager priva di background tecnico specifico possa comunque gestire efficacemente un tema tecnico delicato come il debito tecnico, attraverso ascolto, trasparenza nella comunicazione, e una negoziazione informata con il team e con la direzione, confermando come le competenze trasversali già presentate nella Dispensa 1 di questo percorso siano spesso più decisive della sola competenza tecnica specifica.

## 10. Glossario dei termini chiave

---

Come per le dispense precedenti di questo percorso, riportiamo un glossario di sintesi dei principali termini introdotti.

|                                |                                                                                                                                           |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Sprint</b>                  | Ciclo di lavoro breve e ricorrente, tipicamente di una o due settimane, al termine del quale il team rilascia un incremento verificabile. |
| <b>Frontend</b>                | Parte del software con cui l'utente interagisce direttamente, l'interfaccia visiva.                                                       |
| <b>Backend</b>                 | Infrastruttura non visibile che elabora i dati e la logica sottostante di un software.                                                    |
| <b>User story</b>              | Formato che descrive una funzionalità dal punto di vista del beneficio per un tipo specifico di utente.                                   |
| <b>Criteri di accettazione</b> | Condizioni concrete e verificabili che determinano quando una funzionalità è considerata completa.                                        |
| <b>Debito tecnico</b>          | Insieme delle scelte tecniche non ottimali che nel tempo rallentano progressivamente lo sviluppo futuro.                                  |

## 11. Checklist finale e autoverifica

---

Come per le dispense precedenti, dedichiamo un momento all'autoverifica personale prima di proseguire.

### Domande di autoverifica

1. Sapresti indicare almeno due elementi per costruire fiducia con il team tecnico senza autorità gerarchica?
2. Ricordi che cos'è uno sprint e la differenza tra frontend e backend?
3. Sapresti scrivere la struttura di base di una user story?
4. Ricordi che cos'è il debito tecnico e perché riguarda direttamente il Product Manager?
5. Sapresti spiegare come negoziare uno spazio per la manutenzione nella roadmap?
6. Ricordi perché il design dovrebbe essere coinvolto fin dalle fasi iniziali, non solo dopo la decisione già presa?

### Come procedere

Se le risposte sono arrivate con sicurezza, si può proseguire con la Dispensa 8, dedicata al lancio di una nuova funzionalità sul mercato. In caso di incertezza, si consiglia di rileggere i capitoli corrispondenti, in particolare quelli relativi al debito tecnico e alla scrittura di user story.

## 12. Riferimenti e risorse aggiornate al 2026

---

Per approfondimenti autonomi, si segnalano alcune categorie di risorse utili:

- Pubblicazioni di riferimento sulla collaborazione tra Product Manager e team tecnico;
- Corsi introduttivi sui principi di base dello sviluppo software, pensati specificamente per Product Manager senza background tecnico;
- Comunità professionali di Product Manager, per un confronto su come gestire concretamente il debito tecnico nella propria organizzazione;
- Il confronto diretto con il proprio team tecnico e con i designer coinvolti, la risorsa più preziosa per costruire una comprensione reciproca autentica.

### Chiusura della Dispensa 7

Con questa dispensa prosegue il percorso Product Manager, fornendo gli strumenti per collaborare efficacemente con il team tecnico e per gestire consapevolmente il debito tecnico. La dispensa successiva affronterà il lancio di una nuova funzionalità sul mercato, un momento cruciale che richiede il coordinamento di più funzioni aziendali.

**Marco Consiglio**

Formatore e Course Director — 2026