

Sviluppo e convalida di soluzioni e Prodotto Minimo Vitale

Introduzione

Una volta ottenuto un concetto di business iniziale attraverso tutti gli screening precedenti, e si ha la certezza (in qualsiasi misura sia possibile) che sia degno di investire ulteriormente il proprio tempo e i propri sforzi, verrà raggiunto un punto nel business design in cui si dovrà creare un **prodotto minimo vitale**.

Il cosiddetto Minimum Viable Product (MVP) è una strategia utilizzata per test di mercato rapidi e quantitativi di una proposta di valore, prodotto, servizio o funzionalità. Inizialmente è stato utilizzato per applicazioni web e reso popolare da Eric Ries. Un MVP non è sempre una versione più piccola o più economica del prodotto finale. Spesso una delega più economica consente di testare le ipotesi alla base della proposta di valore in modo più rapido ed economico. **Bisogna chiedersi cosa si vuole veramente imparare e qual è il trucco più economico possibile per provare questa ipotesi.** Di seguito, ci saranno utili approfondimenti sullo sviluppo e il test MVP, per gentile concessione di Dan Olsen ("*The Lean Product Playbook: How to Innovate with Minimum Viable Products and Rapid Customer Feedback*"). Usare le sezioni per facilitare lo sviluppo del proprio MVP.

Specificare l'insieme di funzionalità MVP (Minimum Viable Product)

Se, ormai, si ha **una chiara comprensione della proposta di valore**, il passo successivo nel Lean Product Process è **decidere l'insieme di funzionalità per l' MVP (Minimum Viable Product) del candidato**. Non si inizierà progettando un nuovo prodotto che offra la proposta di valore completa, poiché ciò richiederebbe troppo tempo e sarebbe troppo rischioso. Per il proprio MVP, si può identificare la funzionalità minima richiesta per confermare che si sta andando nella giusta direzione. Si può chiamare un **candidato** MVP invece di **un** MVP perché si basa sulle ipotesi. Si deve comunque verificare con i clienti che si tratti, in effetti, di un prodotto praticabile.

Per analizzare ogni vantaggio nella proposta di valore del prodotto si può fare brainstorming come team per trovare più idee di funzionalità possibili. Prima vengono analizzati i problemi e poi si passa ad analizzare le soluzioni. Mentre il team fa brainstorming, sulla base dei vari suggerimenti ci si può spingere a trovare idee ancora più creative e stravaganti. Quando si finisce il brainstorming, vengono catturate tutte le idee generate dal team e vengono organizzarle in base al vantaggio che offrono. Quindi, per ogni vantaggio, si desidera rivedere e assegnare priorità all'elenco di idee

A questo punto, dovrebbero essere applicate le regole del brainstorming. Andrebbe praticato il pensiero divergente, il che significa cercare di generare quante più idee possibili senza alcun giudizio o valutazione. Ci sarà un sacco di tempo più tardi per il pensiero convergente, dove vengono valutate le idee e vengono scelte le più promettenti.

di funzionalità. È possibile assegnare un punteggio a ogni idea in base al valore previsto per il cliente per determinare una priorità di primo passaggio. L'obiettivo è quello di identificare le prime tre o cinque funzionalità per ogni vantaggio. Non c'è molto valore nel guardare oltre quelle caratteristiche principali in questo momento perché le cose cambieranno molto dopo aver mostrato il proprio prototipo ai clienti.

Storie utente: funzionalità con vantaggi

Le storie utente (utilizzate in Agile development) sono un ottimo modo per scrivere le proprie idee di funzionalità per assicurarsi che il vantaggio del cliente corrispondente rimanga chiaro. Una storia utente è una breve descrizione del vantaggio che la particolare funzionalità deve fornire, incluso a chi è destinato (il cliente di destinazione) e perché il cliente desidera il vantaggio. Le storie utente ben scritte di solito seguono il modello:

Come [tipo di utente],

Voglio [fare qualcosa],

in modo che io possa [beneficio desiderato].

Ecco un esempio di una storia utente che segue questo modello:

Come fotografo professionista,

Voglio caricare facilmente le immagini dalla mia fotocamera al mio sito web,

in modo da poter mostrare rapidamente ai miei clienti le loro foto.

Questo modello è un buon inizio, ma scrivere buone storie utente è un'abilità acquisita. Il leader di pensiero agile Bill Wake ha creato una serie di linee guida per scrivere buone storie utente; per renderli più facili da ricordare, usa l'acronimo **INVEST**:

- **Indipendente:** una buona storia dovrebbe essere indipendente da altre storie. Le storie non dovrebbero sovrapporsi nel concetto e dovrebbero essere implementabili in qualsiasi ordine.
- **Negoziabile:** una buona storia non è un contratto esplicito per le funzionalità. I dettagli su come verrà consegnato il beneficio di una storia dovrebbero essere aperti alla discussione.
- **Prezioso:** una buona storia deve essere preziosa per il cliente.

- **Stimabile:** una buona storia è quella la cui portata può essere ragionevolmente stimata.
- **Piccolo:** le buone storie tendono ad essere di piccola portata. Le storie più grandi avranno maggiore incertezza, quindi dovresti scomporle.
- **Testabile:** una buona storia fornisce informazioni sufficienti per chiarire come testare che la storia sia "fatta" (chiamati *criteri di accettazione*).

Riduzione delle funzionalità

Dopo aver scritto storie utente di alto livello per le funzionalità principali, il passaggio successivo consiste nell'identificare i modi per suddividere ciascuna di esse in parti di funzionalità più piccole, un processo chiamato "**chunking**". **L'obiettivo è trovare modi per ridurre l'ambito e costruire solo i pezzi più preziosi di ogni funzionalità.** Quando a qualcuno viene in mente un'idea caratteristica, ci sono spesso modi creativi per tagliare pezzi meno importanti. Vengono usati deliberatamente il termine "feature chunk" invece di feature per ricordarvi che non dovrete lavorare con elementi di grande portata, ma piuttosto suddividere tali elementi in componenti atomici più piccoli.

Nelle prossime righe viene illustrata l'idea di suddividere una storia utente di alto livello. Se si sta lavorando su un'applicazione di condivisione di foto e si inizia con la storia dell'utente: "Come utente, voglio essere in grado di condividere facilmente le foto con i miei amici in modo che possano goderne". Un modo per suddividere questa storia è dai vari canali che un cliente può utilizzare per condividere foto: Facebook, Twitter, Pinterest, e-mail, messaggi di testo e così via. Ognuno di questi sarebbe un blocco di funzionalità distinto o una storia utente di ambito più piccolo. Potrebbe non essere necessario creare tutti questi canali di condivisione per il tuo MVP. Anche se si è deciso di farlo, è utile suddividere la storia in modo da essere più specifici nella definizione del prodotto, per consentire un ambito più accurato dallo sviluppo e per consentire di assegnare priorità esplicite all'ordine in cui si costruiscono i blocchi. È inoltre possibile limitare l'ambito consentendo all'utente di condividere solo la foto e nient'altro per il server MVP. Potrebbe nascere l'idea di funzionalità aggiuntive lungo la strada, come l'aggiunta di un messaggio opzionale a ogni foto o la possibilità di taggare gli utenti nelle foto. Ognuno di questi sarebbe un pezzo di funzionalità distinto.

Le dimensioni dei lotti più piccole sono migliori

La tattica di scomporre le caratteristiche è coerente con la migliore pratica di produzione Lean di lavorare in lotti di piccole dimensioni. Quando un prodotto viene fabbricato in una linea di fabbrica, la dimensione del lotto è il numero di prodotti su cui si lavora insieme contemporaneamente (in ogni fase del processo di produzione). Il parallelo per lo sviluppo del software è la dimensione delle funzionalità o delle storie

utente da codificare. **Lavorare in lotti di dimensioni ridotte aumenta la velocità perché essi consentono un feedback più rapido, che riduce i rischi e gli sprechi.** Se uno sviluppatore trascorre un mese al computer a sviluppare una funzionalità e poi la mostra al product manager e al designer, c'è una maggiore possibilità che ci sia una disconnessione e che il suo feedback richieda modifiche significative.

Se, invece, lo sviluppatore mostra il suo lavoro al product manager e al designer ogni giorno o *due*, ciò impedisce che si verifichi una grande disconnessione. L'entità del feedback e delle correzioni di rotta sarà molto più piccola e più gestibile, con conseguente riduzione dello spreco di lavoro e maggiore produttività.

Questo consiglio vale anche per i product manager e i designer che mostrano il loro prodotto di lavoro (ad esempio, storie utente e wireframe) ai loro compagni di squadra. Il vantaggio di lavorare in lotti di piccole dimensioni si applica anche al feedback dei clienti. Più a lungo si lavora su un prodotto senza ottenere il feedback dei clienti, più si rischia una disconnessione importante che successivamente richiede una rilavorazione significativa.

Decidere il Prodotto Minimo Vitale del candidato

Una volta terminata la suddivisione, l'ambito e la definizione delle priorità, è possibile creare una semplice routine che elenca i vantaggi derivanti dalla proposta di valore e che elenca per ogni vantaggio, le idee di funzionalità principali suddivise in blocchi.

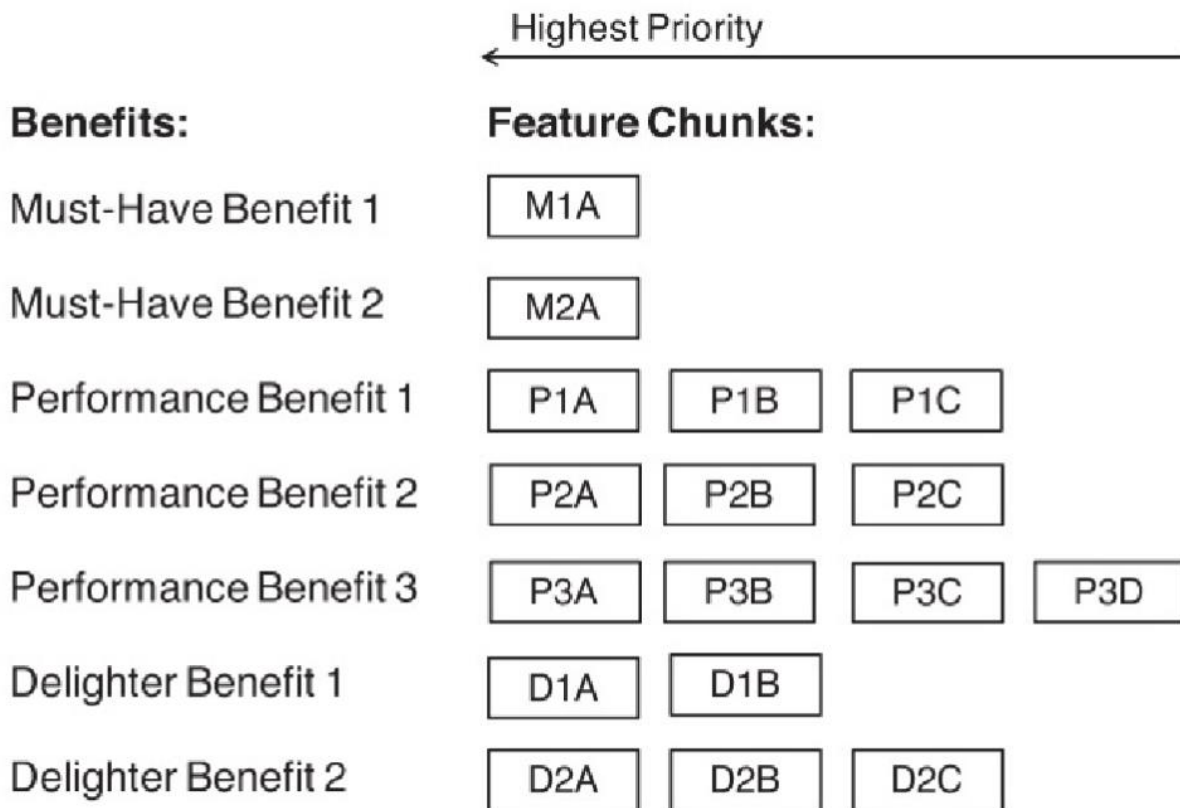


FIGURA 1. ELENCO DEI BLOCCHI DI FUNZIONALITÀ PRIORITARI PER OGNI VANTAGGIO

Nella Figura 1 sono elencati i blocchi di funzionalità principali per ciascun vantaggio in ordine di priorità, con priorità più alta a sinistra. Piuttosto che nominare vantaggi specifici o parti di funzionalità, abbiamo intenzionalmente assegnato loro nomi generici in modo che ognuno possa immaginare più facilmente di sostituirli con ciò che sarebbe rilevante per il proprio prodotto. "M1A" significa caratteristica pezzo A per must-have 1. "P2B" indica il feature chunk B per il vantaggio di prestazioni 2 e "D2C" indica il feature chunk C per il vantaggio del delighter 2. Nel compilare una griglia simile per il proprio prodotto, si utilizzeranno invece le etichette specifiche per i propri vantaggi e i blocchi di funzionalità.

Dopo aver organizzato l'elenco dei blocchi di funzionalità in base al vantaggio e aver dato loro la priorità, è il momento di iniziare a prendere alcune decisioni difficili. **Bisogna decidere l'insieme minimo di funzionalità che risuonerà con il target dei clienti.** In basso nella colonna più a sinistra ci sono dei blocchi di funzionalità e bisogna determinare quali debbano essere nel proprio MVP del candidato. Contemporaneamente si fa riferimento alla proposta di valore del prodotto. Per cominciare, il candidato MVP deve avere tutti i must-have che sono stati identificati.

Successivamente, bisogna concentrare l'attenzione sul principale vantaggio in termini di prestazioni che si intende utilizzare per battere la concorrenza. È necessario selezionare l'insieme di blocchi di funzionalità per questo vantaggio che possa fornire abbastanza dati per consentire ai clienti di vedere la differenza nel prodotto.

Anche i delighters fanno parte della tua differenziazione. Si deve includere il top delighter nell' MVP del candidato. Ciò potrebbe non essere necessario se si dispone di un vantaggio molto elevato in termini di prestazioni. L'obiettivo è assicurarsi che l'MVP del candidato includa *qualcosa* che i clienti trovino superiore ad altri prodotti e, idealmente, unico. I blocchi di funzionalità che si ritiene debbano essere presenti nell'MVP del candidato rimarranno nella colonna più a sinistra, che è possibile etichettare come "v1", come illustrato nella Figura 2, mentre gli altri vengono spostati verso destra. È possibile creare una tabella di marcia preliminare del prodotto continuando questo processo e creando colonne per ogni versione futura con ogni colonna contenente i blocchi di funzionalità che si intende aggiungere.

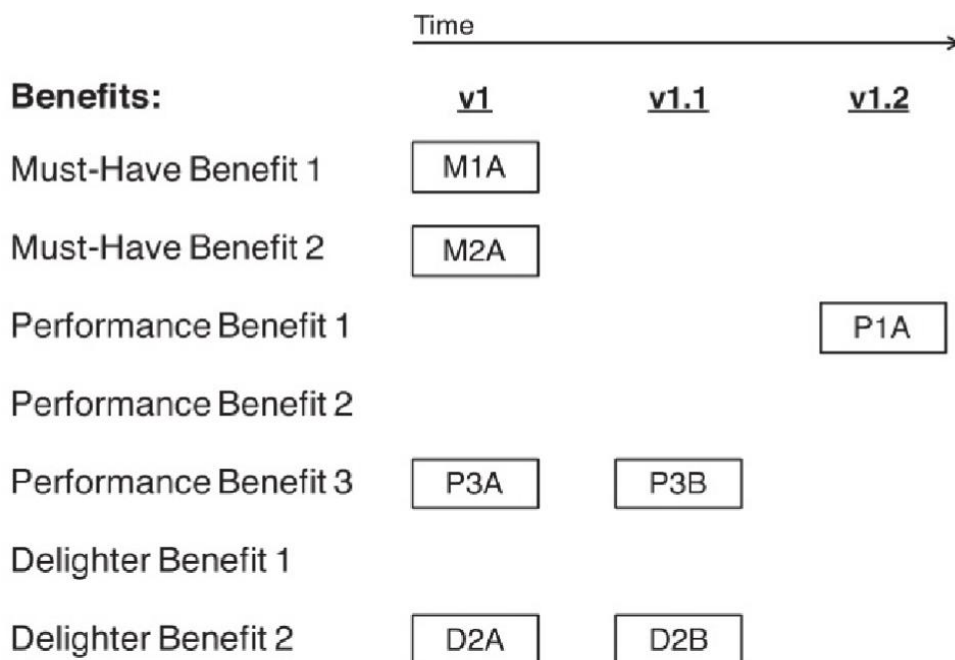


FIGURA 2. DECIDERE QUALI BLOCCHI DI FUNZIONALITÀ SONO PRESENTI NEL CANDIDATO MVP

Dal momento che si prevede di essere il migliore nel vantaggio delle prestazioni 3, viene incluso il blocco di funzionalità con priorità più alta, P3A, nell' MVP del candidato. Prevedi anche di differenziarti con il differenziatore 2, in modo da includere il blocco di funzionalità D2A nell' MVP del candidato. L'MVP del candidato ha anche i due must-have.

Guardando alla tua prossima versione, v1.1, prevedi di investire ulteriormente nel vantaggio delle prestazioni 3 e nel delighter 2 con blocchi di funzionalità P3B e D2B, rispettivamente. Nella versione successiva, v1.2, si prevede di iniziare ad affrontare il vantaggio in termini di prestazioni 1 con il blocco di funzionalità con priorità più alta P1A.

Non è consigliabile pianificare più di una o due versioni secondarie all'inizio, poiché molte cose sono suscettibili di cambiare quando si mostra l'MVP del candidato ai clienti per la prima volta. Col tempo si vedrà che alcune ipotesi non erano del tutto giuste e ne saranno inventate di nuove. Tutto questo potrebbe far cambiare idea su quale vantaggio sia più importante o trovare idee per nuove funzionalità per affrontare gli stessi vantaggi. Quindi bisogna essere pronti a mettere da parte i propri piani provvisori per lasciare spazio a quelli nuovi basati su ciò che si impara dai clienti. Tuttavia, potrebbero essere necessari due o tre blocchi di funzionalità per un determinato vantaggio, a seconda della propria situazione e di quanto sono piccoli i pezzi. L'idea è sempre la stessa: scegliere quali pezzi di funzionalità devono essere in quella colonna più a sinistra, che corrisponde al MVP del candidato.

Facciamo un passo indietro e riflettiamo. A questo punto del Lean Product Process, un bel po' di lavoro è stato fatto:

- Ipotesi formate sul target dei clienti
- Ipotesi formate sui loro bisogni
- Articolazione della proposta di valore che si intende perseguire in modo che il prodotto sia migliore e diverso
- Identificazione delle idee di funzionalità principali che si ritiene possano soddisfare tali esigenze e suddivisione in blocchi più piccoli
- Assegnazione della priorità ai blocchi di funzionalità in base al ROI
- Selezione di un insieme di quei blocchi di funzionalità per l'MVP del candidato, che i clienti troveranno preziosi

Nonostante un pensiero molto articolato l'MVP è ancora solo un candidato, un fascio di ipotesi correlate. Bisogna ottenere il feedback dei clienti sull'MVP del candidato per testare tali ipotesi. Ma prima di poter eseguire il test, è necessario creare una rappresentazione dello spazio della soluzione dell'MVP del candidato che è possibile mostrare ai clienti, che è il passaggio successivo nel Lean Product Process.

Creazione del tuo MVP: cos'è (e non è) un MVP?

C'è stato un vivace dibattito su ciò che si qualifica come MVP. Alcune persone sostengono con veemenza che una landing page è un MVP valido. Altri dicono che non lo è, insistendo sul fatto che un MVP deve essere un prodotto reale e funzionante o almeno un prototipo interattivo. Il modo per risolvere questa dicotomia è rendersi conto che questi sono tutti metodi per *testare* le ipotesi alla base del tuo MVP. Usando il termine "test MVP" invece di MVP, il dibattito scompare. Ciò consente una terminologia più precisa riservando l'uso di MVP per i prodotti reali.

Molte persone interpretano erroneamente il termine MVP ponendo troppa enfasi sulla parola *minimo*. Usano questo come scusa per creare un MVP parziale che ha troppe poche funzionalità per essere considerato fattibile da un cliente. Altri usano "minimum" per razionalizzare un'esperienza utente scadente o un prodotto difettoso.

Mentre è vero che un MVP è deliberatamente limitato nell'ambito rispetto all'intera proposta di valore, ciò che rilasci ai clienti deve essere al di sopra di una certa barra per creare valore per loro.

Il diagramma nella Figura 3 illustra la differenza tra questo modo errato di interpretare MVP e l'interpretazione corretta.

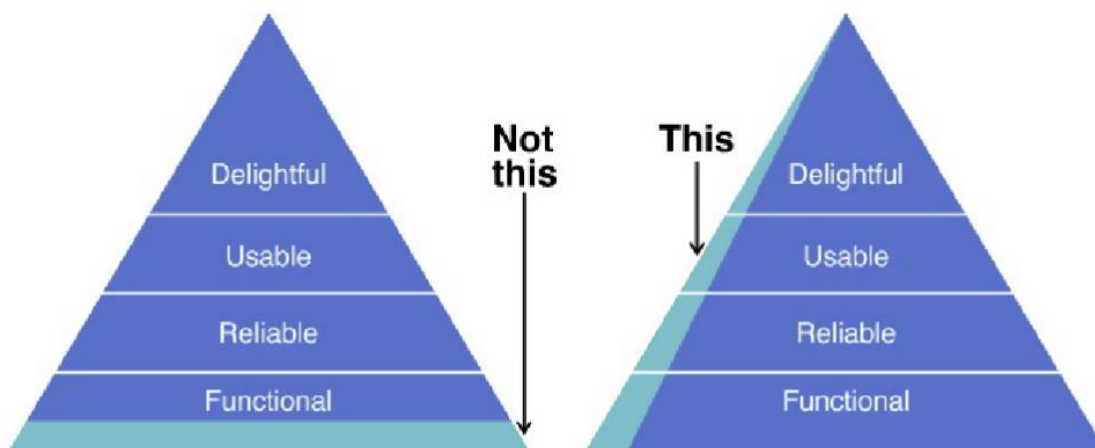


FIGURA 3. CREAZIONE DI UN MVP

Simile alla gerarchia delle esigenze degli utenti web, questa cifra separa gli aspetti distinti di un prodotto. In questo caso, **una piramide di quattro livelli gerarchici viene utilizzata per descrivere gli attributi di un prodotto: funzionale, affidabile, utilizzabile e piacevole**. La piramide a sinistra illustra l'idea sbagliata che un MVP sia solo un prodotto con funzionalità limitate e che affidabilità, usabilità e piacere possano essere ignorati. Invece, la piramide a destra mostra che mentre un MVP ha funzionalità limitate, dovrebbe essere "completo" affrontando questi tre attributi di livello superiore.

Riepilogo - Test MVP

Infine, dopo aver fatto tutto il lavoro, ora è il momento di testare il proprio MVP. Per una conclusione approfondita della nostra sessione dedicata allo sviluppo e al test degli MVP, va controllato la struttura del test dell'appendice (per gentile concessione di Strategyzer) e trovate le tecniche di test più appropriate che si possono applicare durante la fase di ricerca e progettazione della propria idea di business.

È bene parlarne nel proprio team e scegliere da dove / come iniziare di conseguenza.

Ulteriori letture e informazioni:

- [\(PDF\) Prodotto minimo vitale o prodotto multifaccetta? Il ruolo di MVP nelle startup software \(researchgate.net\)](#)
- [Sviluppare la soluzione \(iitoolkit.com\)](#)
- [Validazione della soluzione e costruzione dell'MVP - Scaleapse](#)
- [Qual è la fase MVP in una startup? | Londra Business Advice \(hwca.com\)](#)

Fonti e riferimenti:

AG, S. (senza data) Test del modello di business, Resource Library. Disponibile all'indirizzo: <https://www.strategyzer.com/assets/resources> (accesso: 7 novembre 2022).

Miglioramento del miglioramento (senza data) Sviluppare la soluzione. Disponibile all'indirizzo:

<https://www.iitoolkit.com/process/solution.html#:~:text=Developing%20the%20solution%20has%20direct,evaluate%20potential%20ideas%20for%20improvement> (accesso: 7 novembre 2022).

Pansini, S. (2022) Validazione della soluzione e costruzione dell'MVP, Scaleapse. Disponibile all'indirizzo: <https://www.scaleapse.com/validation-of-the-solution-and-construction-of-the-mvp/> (accesso: 7 novembre 2022).

Umbraco.Web.PublishedModels.TeamMember (2018) Qual è la fase MVP in una startup?: London Business Advice, Haines Watts Group. Gruppo Haines Watts. Disponibile all'indirizzo: <https://www.hwca.com/accountants-london/opinion/mvp-stage-tech%20startup/#:~:text=MVP%20is%20an%20abbreviation%20for,basic%20principles%20is%20relatively%20simple> (accesso: 7 novembre 2022).

Burns, P. (2018). Creazione di nuove imprese. Macmillan Education Regno Unito. <https://doi.org/10.1057/978-1-352-00051-1>

Lenarduzzi, V. (2017) MVP spiegato: A systematic mapping study on the definitions of minimal viable product, 2016 42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA). Disponibile [all'indirizzo: https://www.academia.edu/33157207/MVP_Explained_A_Systematic_Mapping_Study_on_the_Definitions_of_Minimal_Viable_Product](https://www.academia.edu/33157207/MVP_Explained_A_Systematic_Mapping_Study_on_the_Definitions_of_Minimal_Viable_Product) (accesso: 7 novembre 2022)

Nguyen Duc, Anh & Abrahamsson, Pekka. (2016). Prodotto minimo vitale o prodotto a sfaccettature multiple? Il ruolo di MVP nelle startup software. 118-130. 10.1007/978-3-319-33515-5_10.