Varad Koppar
PHOTOGRAPHY

Product Management Fundamentals

Types of Companies

1. Business Models

B2C (Business-to-Consumer)
Example: Amazon, Netflix, Apple (iPhones).

B2B (Business-to-Business)
Example: Salesforce (CRM software for businesses), AWS (cloud computing services).

2. Service Delivery Models

On-Premise
Example: Banks running their own data centers, SAP ERP installed on private servers.

SaaS (Software as a Service) - Cloud-Based
Example: Google Workspace, Dropbox, Zoom.

3. IT & Software-Specific Business Models

ISV (Independent Software Vendor)
Example: Adobe (Photoshop, Illustrator), Atlassian (Jira, Confluence).

SI (System Integrator)
Example: Accenture, Infosys, Capgemini.

MSP (Managed Service Provider)
Example: Rackspace, IBM Managed Services.

Which Companies Require Product Managers?

Product Managers (PMs) play a crucial role in B2C, B2B, SaaS, ISVs, and some MSPs.

B2C & B2B companies need PMs to define user needs, prioritize features, and ensure products align with business goals.

SaaS & ISVs heavily rely on PMs to manage product roadmaps, enhance user experience, and drive innovation in cloud-based solutions.

System Integrators (SI) typically focus on service delivery rather than product innovation, so PM roles are less common.

Managed Service Providers (MSPs) may require PMs when they develop proprietary platforms or software tools.

PMs are essential wherever product development, strategy, and customer experience drive business success.

What is a Product?

A product is anything that creates value for users—whether it is a physical good, digital software, or a service. Products solve specific problems and can be priced based on the value they provide, such as cost savings, convenience, or efficiency.

Are Services Products?
Yes, services can be considered products because they provide value, solve problems, and can be priced. For example, Spotify (a digital service) and Coca-Cola (a physical product) both deliver value, making them products in different forms.

Are Brands Products?
A brand itself is not a product, but it enhances a product's value by influencing perception, trust, and differentiation in the market. For example, Apple's iPhone is a product, but Apple as a brand increases its perceived value.

Types of Products

Disruptive Innovative Products

- Products that transform industries by introducing groundbreaking solutions.
- Example: Tesla (electric vehicles), Netflix (streaming services).

Complementary Products

- Products that add value to or enhance another product.
- Example: AirPods (complementary to iPhones), Game controllers (for gaming consoles).

Substitute Products

- Products that serve the same purpose and can replace each other.
- Example: Tea vs. Coffee, Uber vs. Public Transport.

What is Product Management ?

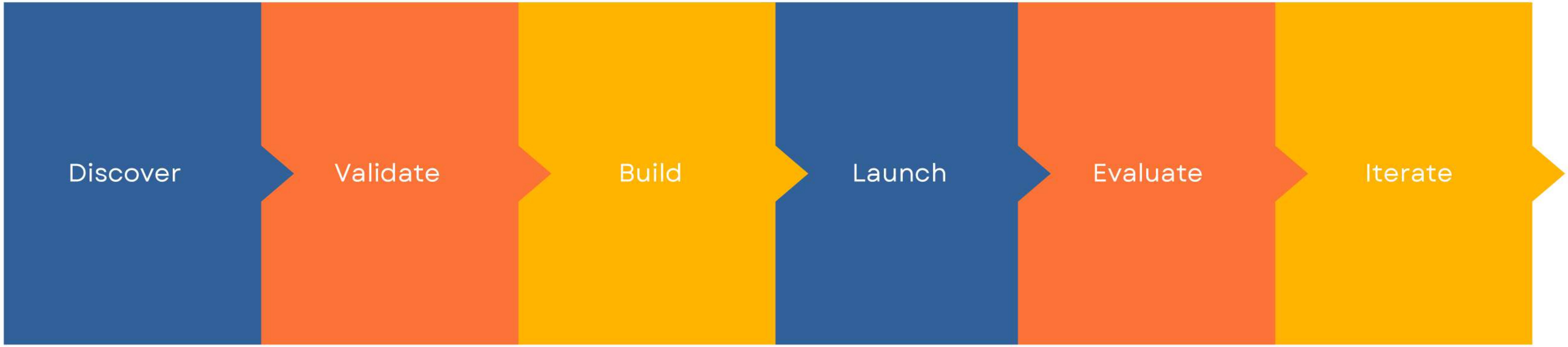
What is Product Management?

Product Management is the strategic process of guiding a product's development, launch, and success. It involves understanding customer needs, defining features, prioritizing development, and ensuring business objectives align with user value.

A Product Manager (PM) collaborates with cross-functional teams like engineering, design, marketing, and sales to build and improve products. The role focuses on:

- Identifying market opportunities
- Defining product vision & strategy
- Prioritizing features & managing roadmaps
- Gathering customer feedback for continuous improvement

The Product Management Life Cycle



Discover: Understand customer needs and problems.

Validate: Test and prioritize the right solution.

Build: Develop and deliver continuous value.

Launch: Bring the product to market.

Evaluate: Analyze product performance.

Iterate: Improve the product based on feedback.

Varad Koppar
PHOTOGRAPHY

Product Lifecycle Growth Stage & Beyond

Product Lifecycle: Growth Stage & Beyond

After achieving Product-Market Fit (PMF) and launching the right version of the product successfully, the next goal is growth—acquiring new users and expanding market reach.

Growth Stage

Building the Right Team – Expanding the product, engineering, sales, and marketing teams to scale operations efficiently.

Creating Capacity – Ensuring infrastructure, customer support, and operational processes can handle increasing demand.

Marketing to Target Users – Identifying the right audience, refining the Unique Selling Proposition (USP), and choosing the right acquisition channels (SEO, ads, partnerships, etc.).

Pricing Strategy – Adjusting pricing to balance affordability, profitability, and competitive positioning.

Product Lifecycle Growth Stage & Beyond

Maturity Stage

Improving Retention – Enhancing user experience, customer support, and engagement strategies to keep existing users.

Keeping Up with Competition – Innovating and differentiating against market challengers.

Upselling & Expanding Offerings – Introducing premium features, additional services, or product variations to maximize revenue.

Decline Stage

Eventually, every product faces a decline due to market saturation, competition, or changing trends. The key focus areas here include:

Pivoting or Reinventing the Product – Introducing new features, markets, or business models.

Managing Costs – Reducing overhead while maintaining product quality.

If the product is no longer viable, the ultimate truth is to phase it out and shift resources elsewhere, especially when competition is providing more value, better features, or lower costs, making the product obsolete.

Product Team Goals

A product team focuses on several key objectives to ensure the success and growth of a product:

Retention – Keeping existing users engaged and ensuring they continue using the product over time.

User Acquisition – Attracting new customers through effective marketing, referrals, and delivering product value.

Customer Satisfaction – Enhancing the user experience, reducing friction, and addressing pain points to improve loyalty.

Getting Prioritization Right – Focusing on the most impactful features, improvements, and strategic initiatives to drive both business and user success.

What is Vision ?

Vision is the ability to think, plan, and shape the future with imagination and intent. It is driven by experiences, learning, and mistakes, shaping how individuals and teams perceive opportunities and challenges.

In product management, vision is born from the desire to solve a problem—people build products because they see gaps, inefficiencies, or unmet needs in the world. A strong vision inspires action, aligns teams, and provides direction for long-term success. It is not just about having an idea but about having the conviction and clarity to work on a meaningful problem and bring impactful solutions to life.
Vision is also like the brain of a product—it provides clarity, direction, and purpose. It is bold, pushing boundaries beyond the present. A strong vision is relatable, addressing real-world problems that resonate with people. It must be ambitious, setting high aspirations while remaining inspiring to teams and stakeholders. Above all, a vision should be navigable, offering a clear path to execution while adapting to challenges along the way.

A great vision doesn't just imagine the future—it drives action and rallies people around a purpose to build something impactful.

Hierarchy

Vision The long-term purpose and aspiration of the product. Why are we building this?

Goals Measurable objectives aligned with the vision. What do we want to achieve?

Product Strategy The plan to reach the goals. How will we achieve them?

Initiatives High-level efforts or projects supporting the strategy. What key areas will we focus on?

Features Specific functionalities or improvements within initiatives. What will we build?

Epics Large bodies of work grouping related features. What broader tasks need to be done?

User Stories Detailed descriptions of user needs and interactions. How does this benefit the user?

Tasks Actionable work items for development teams. What steps are needed to complete it?

MVP Minimum Viable Product & Related Concepts

What is an MVP?

A Minimum Viable Product (MVP) is the most basic version of a product that allows a team to test its viability with real users while investing minimal time and resources.
The concept was introduced by Eric Ries in The Lean Startup, where he defines MVP as:
“The version of a new product which allows a team to collect the maximum amount of validated learning about customers with the least effort.”
MVP focuses on prioritizing only essential features to test whether the product is worth further investment.

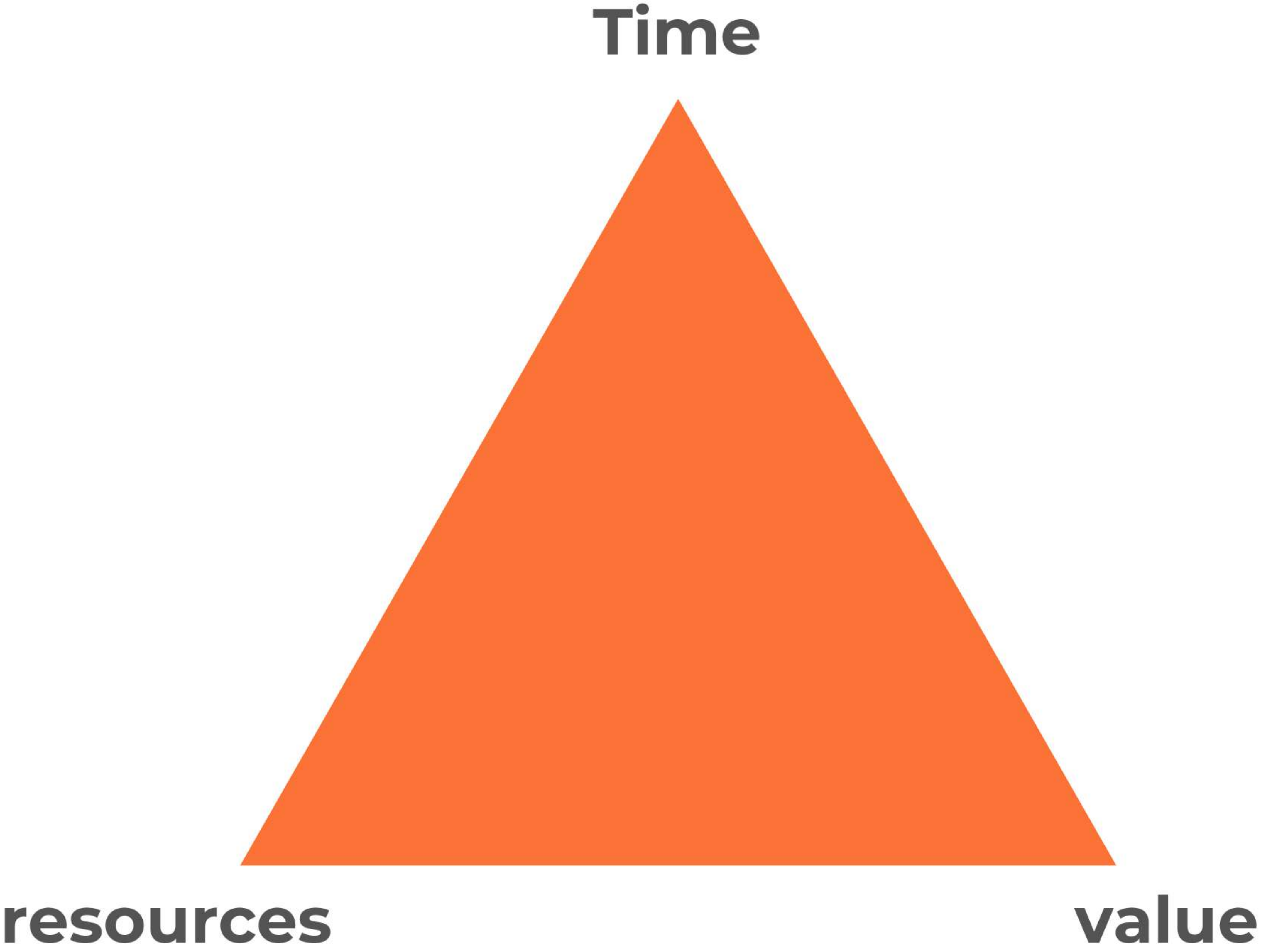
Related Concepts: Understanding the Development Stages

POC (Proof of Concept) – Is it technically possible?
A small experiment to test if the core idea or technology can work.

Prototype – How will users interact with it?
A visual or interactive model to showcase the product’s design and flow.

MVP (Minimum Viable Product) – What do users think?
A functional product with minimal features to test market demand.

MMP (Minimum Marketable Product) – First marketed version of the product
A more polished version of the MVP, ready for wider adoption and monetization.



Time

resources

value

MVP Testing: Key Questions for Validation

A good way to test an MVP (Minimum Viable Product) is to ask the following critical questions:

Does it solve the core problem it is focused on?

The MVP should directly address a real user need or pain point.

Can we cut down the scope further without impacting the core value?

Eliminating non-essential features ensures faster development and testing.

Have we reduced the scope so much that the product is now bare-bones?

If the product lacks key functionality, it may no longer be useful to users.

The intention behind these questions is to balance time, resources, and value while ensuring the product delivers its key value proposition effectively.

Build-Measure-Learn (BML) & MVP Framework

BML: Build, Measure, Learn

The Build-Measure-Learn (BML) loop is a core principle of the Lean Startup methodology, ensuring efficient product iteration and validation.

- Why? To validate ideas quickly and reduce waste.
- How? By continuously testing, learning, and improving.
- What? A cycle where you build an MVP, measure its success, and learn from user feedback.

Learn: Understanding User Behavior

- Who is using the product? Identifying early adopters.
- How often are they using it? Evaluating engagement.
- How do they reference it? Understanding user perception.
- What does feedback suggest? Identifying improvements.

Build-Measure-Learn (BML) & MVP Framework

Advantages of an MVP

Testing Hypotheses – Validates whether the core idea solves a real problem.

Longer Vision in Core Function – Focuses on essential features without distractions.

Identifies Necessities – Helps distinguish must-have vs. nice-to-have features.

Fails Fast – Saves time and resources by learning early.

Real-World User Data – Provides insights into actual user behavior.

Challenges with MVP

MVP may reveal the idea isn't worth investing in.

Cutting feature scope too aggressively – Removing too much may make the product ineffective.

Forgetting that MVP ≠ Final Product – The MVP is a test, not a fully polished version.

5 Steps to Build an MVP

Develop Your Hypothesis – If we provide users with X, then Y will happen.

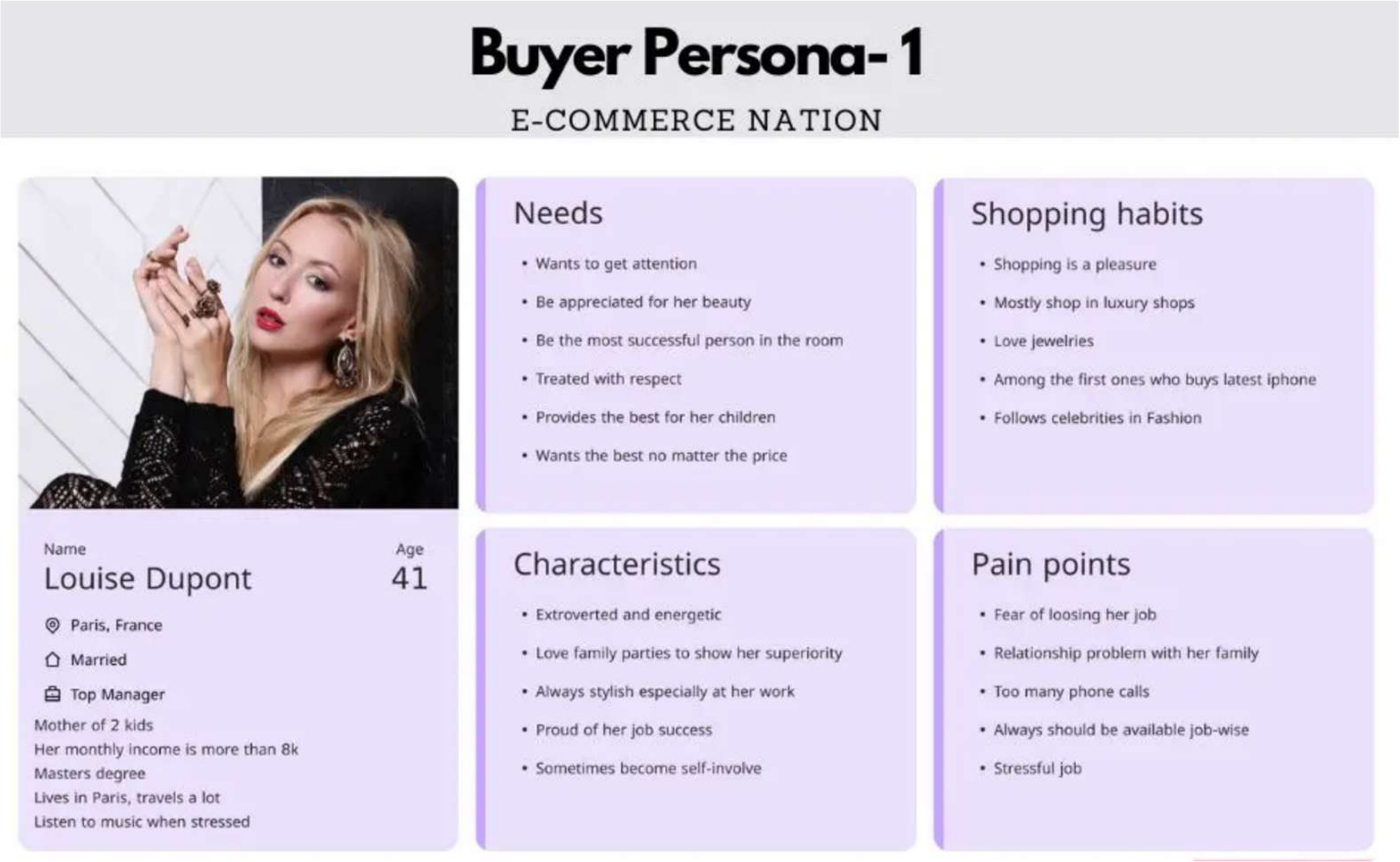
Identify Target Users – Define the ideal customer profile.

Gauge Market Opportunity – Assess demand and competition.

Evaluate Technical Feasibility – Ensure it can be built efficiently.

Implement BML (Build-Measure-Learn) – Develop, test, and iterate based on user feedback.

Persona Example



Buyer Persona- 1

E-COMMERCE NATION

Name
Louise Dupont

Age
41

- Paris, France
- Married
- Top Manager

Mother of 2 kids
Her monthly income is more than 8k
Masters degree
Lives in Paris, travels a lot
Listen to music when stressed

Needs

- Wants to get attention
- Be appreciated for her beauty
- Be the most successful person in the room
- Treated with respect
- Provides the best for her children
- Wants the best no matter the price

Shopping habits

- Shopping is a pleasure
- Mostly shop in luxury shops
- Love jewelries
- Among the first ones who buys latest iphone
- Follows celebrities in Fashion

Characteristics

- Extroverted and energetic
- Love family parties to show her superiority
- Always stylish especially at her work
- Proud of her job success
- Sometimes become self-involve

Pain points

- Fear of loosing her job
- Relationship problem with her family
- Too many phone calls
- Always should be available job-wise
- Stressful job

A buyer persona helps understand user needs by analyzing their shopping habits, demographics, characteristics, and pain points. Each industry or product may require different personas, including technical users, consumers, or business clients.

To create your own, search online for persona templates and customize them based on your specific audience and industry needs. Aim to define at least 5 to 7 different personas for a well-rounded understanding.

What is Product-Market Fit (PMF)?

Product-Market Fit (PMF) means your product effectively solves a real problem for a clearly defined audience, and people love it, use it, and recommend it consistently.

Why Do You Need PMF?

Without PMF, a product may struggle with user adoption, retention, and growth. Achieving PMF ensures that:

- You are solving the right problem for the right audience.
- Customers actively use and love the product.
- There is strong demand and organic growth (word-of-mouth, referrals).

How to Achieve PMF?

1. Understand the Problem

Clearly define the problem you're solving.

Assess if the current solutions in the market are inadequate.

Ensure your product addresses a real user pain point.

What is Product-Market Fit (PMF)?

2. Define a Strong Value Proposition

- Provide a better user experience than competitors.
- Ensure your solution is superior in terms of efficiency, cost, or convenience.
- Make sure your value proposition is clear and compelling to users.

3. Research the Market

- Identify potential users who deeply understand the problem.
- Analyze competitors and their existing customer base.
- Engage in industry conferences, product reviews, and user communities to find insights.
- Study search trends to understand what people are looking for.
- Talk to real users and empathize with their needs.

4. Identify & Engage with Customers

- Conduct customer interviews to understand pain points.
- Observe how they interact with the UI and navigate the product.
- Gather both quantitative (data-driven) and qualitative (user feedback) insights.

Testing for PMF

Testing Product-Market Fit is crucial to see how users respond to your product and whether they genuinely need and love it.

Key Questions to Test PMF:

- Who is using the MVP? (Early adopters, target users)
- How many people are using it? (Engagement and retention metrics)
- How are they using it? (Core features vs. secondary features)
- Are they recommending it to others? (Organic growth & referrals)
- How active are the users? (Daily, weekly, monthly retention)
- What feedback are users providing? (Pain points, feature requests, complaints)
- What is preventing non-users from adopting the product? (Barriers to entry, usability issues, pricing concerns)
- Are users willing to pay for it? (If it's a paid product, are they converting? If free, do they see enough value?)
- How does user behavior compare to competitors? (Are they switching from another product? Why?)

What is a Product Roadmap ?

A Product Roadmap is a strategic representation of a company's product vision and development plans for the next 6, 12, or 18 months. It can be documented in various formats, such as presentations, documents, or visual tools, and serves as a communication tool to align stakeholders.

Why is a Product Roadmap Important?

- Communicates what's happening in the product team and what will be built in the coming months.
- Fosters transparency, ensuring alignment across teams.
- Inspires and empowers teams, providing a clear direction.
- Creates accountability while allowing flexibility to adapt to changes.

Factors That Influence the Roadmap

A roadmap is not static; it evolves based on:

- Vision & Goals – Aligning with the company's long-term objectives.
- Internal Feedback – Insights from sales, support, and leadership.
- Competitors & Market Trends – Staying ahead of industry shifts.

What is a Product Roadmap ?

Each stakeholder group uses the roadmap differently:

Why is a Product Roadmap Important?

- Customers - Understand upcoming features and improvements.
- Investors – Gain confidence in the company's direction.
- Engineering Team – Plan development efforts.
- Sales Team – Align sales pitches with upcoming features.
- Marketing Team – Prepare campaigns based on feature launches.

Building a Product Roadmap

Building a roadmap is not an overnight task—it requires collaboration and is a live document that needs adjustments over time.

Types of Roadmaps:

- Theme-Based Roadmaps – Focus on improving a specific area, e.g., "Enhancing Customer Onboarding" or "Redesigning Signup Flow."
- Objective-Based Roadmaps – Built around business goals, e.g., "Increase revenue by 20%."

What is a Product Roadmap ?

Prioritizing Ideas for the Roadmap

When deciding what to build, ask:

- Is this urgent?
- Will customers pay for it?
- Does it align with the company's vision?

Frameworks for Prioritization

RICE Framework (Reach, Impact, Confidence, Effort) Helps score and prioritize initiatives based on value vs. effort.

Kano Model (Basic, Performance, Delight Features) Categorizes features based on customer satisfaction levels.

MoSCoW Method (Must-Have, Should-Have, Could-Have, Won't-Have) Prioritizes features based on necessity.

Story Mapping – Mapping out the user journey to visualize feature dependencies and importance.

What is a Product Roadmap ?

Product Roadmaps Are Flexible

A product roadmap is subject to change—as the business environment evolves, priorities shift. The roadmap should be adaptive and iterative to reflect new challenges and opportunities.

Communicating the Roadmap:

Be clear on priorities and reasoning.

Set an agenda and allow at least 20 minutes for Q&A.

Present to customers to build trust and commitment.

Use formats like Now-Next-Later to provide flexibility and clarity.

Share with internal teams & customers to maintain transparency.

Avoiding Common Roadmap Pitfalls:

Over-promising features that may change later.

Ignoring customer feedback when shaping priorities.

Treating the roadmap as a fixed plan instead of a flexible strategy.

Understanding Epics & User Stories

What is an Epic?

An epic is a logical collection of related user stories that fit together under a broader theme or functionality. It represents a larger feature or initiative that is broken down into smaller, manageable user stories.

Does it make sense to have an epic for every release?

Not necessarily. Some releases may contain multiple small epics, while others may focus on a single large one.

Do you need to release all user stories in an epic together?

No, user stories within an epic can be released incrementally as they become ready.

What is a User Story?

A user story defines a single interaction between a user and the product. It is an independent unit of work that can be built, tested, and released on its own.

Understanding Epics & User Stories

Why Do We Need User Stories?

- Achieve product goals by breaking down complex features.
- Prioritize development based on impact and feasibility.
- Align teams (PMs, engineers, designers, QA) toward a common goal.
- Estimate effort required for completion.
- Track progress toward feature completion.
- Enable cross-functional collaboration by fitting all puzzle pieces together.

Who Uses & Writes User Stories?

- PMs – Translate ideas into actionable tasks.
- Developers & QA – Use as a single source of truth for implementation & testing.
- Engineering Managers & Designers – Understand how the feature interacts with the user experience.

Anyone can write user stories—developers, designers, co-founders—if they have:
Empathy for users
Understanding of business goals & competitive landscape
Knowledge of the product vision & strategy

Understanding Epics & User Stories

4 Key Elements of a User Story:

- User-Driven Persona – Who is the user?
- Outcome-Oriented – What is the purpose?
- Acceptance Criteria – What must be true for the story to be considered done?
- Definition of Done – Has the story been tested & validated by PM, Designer, QA, and Developer?

User Story Mapping

User story mapping visualizes how users interact with a product. It helps prioritize and organize development based on the user journey.

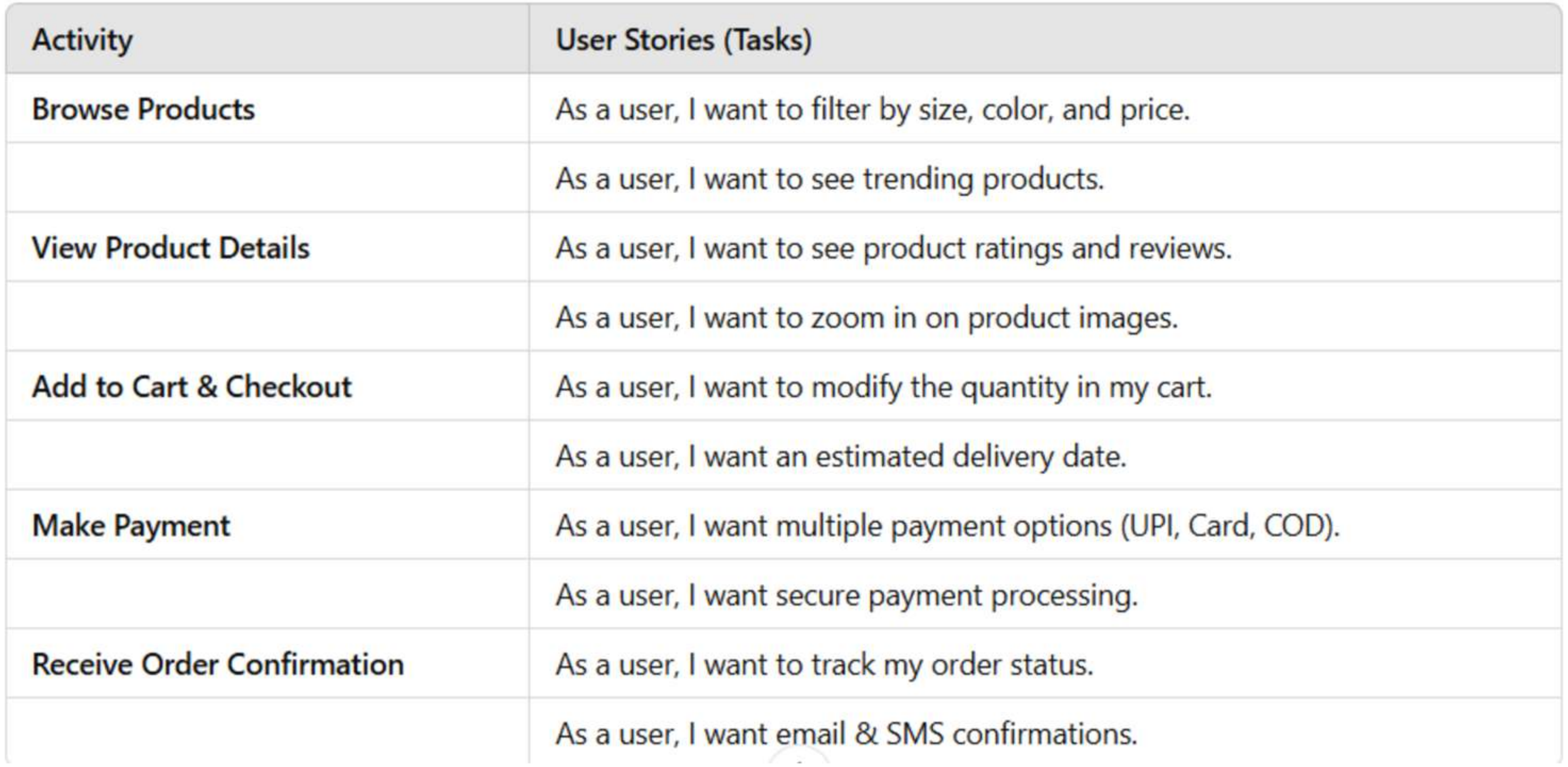
Example (Buying a T-Shirt in a Physical Store vs. Online)
Without tech :

- Go to a store. find parking
- Browse shirts.
- Choose size, color, and style.
- Make payment.

With tech (e-commerce):

1. Open website/app.
2. Search for shirts.
3. Filter options (color, size, price).
4. Add to cart & checkout.
5. Complete payment.

User Story Map Example – E-commerce Shopping Experience



Activity	User Stories (Tasks)
Browse Products	As a user, I want to filter by size, color, and price.
	As a user, I want to see trending products.
View Product Details	As a user, I want to see product ratings and reviews.
	As a user, I want to zoom in on product images.
Add to Cart & Checkout	As a user, I want to modify the quantity in my cart.
	As a user, I want an estimated delivery date.
Make Payment	As a user, I want multiple payment options (UPI, Card, COD).
	As a user, I want secure payment processing.
Receive Order Confirmation	As a user, I want to track my order status.
	As a user, I want email & SMS confirmations.

A User Story Map helps Product Managers prioritize features and align teams while guiding Designers in understanding user interactions for a seamless experience. It ensures development is driven by business goals, user needs, and technical feasibility.

Why this works?

Includes PMs (feature prioritization & team alignment), Designers (user journey & experience design), and Developers (task breakdown & implementation).

Highlights the collaborative nature of user story mapping.

Keeps the focus on strategy, execution, and user needs.

Scrum Process Explained

Scrum Process

Scrum is a widely used framework for building products iteratively and incrementally. It enables companies to deliver fast product increments while adapting to changes in design, architecture, and requirements.

Why is Scrum Popular?

- Fast iterations – Allows quick product improvements.
- Flexibility – Adapts to changing business and user needs.
- Transparency – Ensures clarity among all team members.

Who is Part of a Scrum Team?

Product Owner (or Product Manager) – Represents the business and prioritizes the backlog.

Scrum Master – Facilitates the Scrum process and removes blockers.

Developers – Work on product development (titles don't matter in Scrum).

Scrum Basics: Key Components

- Sprint – A 1 to 4-week cycle to build and deliver increments.
- Product Backlog – A list of features, fixes, and tasks needed to build the product.
- Sprint Backlog – A subset of prioritized tasks for the current sprint.
- Sprint Planning – The team decides what to build in the sprint.
- **Daily Scrum (Standup) – Quick meeting to discuss:**

What was done yesterday? What is the focus today? Any blockers?

- Sprint Review – Team presents completed work to stakeholders.
- **Sprint Retrospective – Team reflects on:**

What went well? What didn't go well? What can be improved?

Advantages of Scrum

Transparency – No ambiguity in workflow.

Incremental Development – Delivers value faster.

Flexibility – Adjusts to changing priorities.

Focus on Necessity – Documentation is lightweight and practical.

Kanban (Board | Principles | Process)

Kanban is a visual workflow management system that focuses on improving work efficiency through incremental progress. It is structured around Kanban boards and cards to track tasks in real-time.

Key Kanban Elements

Kanban Board – Displays tasks visually in columns:
Prioritized Backlog → Development → Testing → Done

Kanban Cards – Represent individual work items with details like:
Title, Story Number, Due Date, Work Type, Platform, Assigned To

Work in Progress (WIP) Limits – Restricts the number of tasks in progress to maintain focus and efficiency.

Swimlanes – Categorizes work types (e.g., Bugs, Features, Enhancements).

Color Coding – Differentiates priority levels or task types.

Kanban Metrics

Lead Time – Time from request to completion.

Cycle Time – Time taken to complete a single task.

Throughput – Number of tasks completed in a given timeframe.

Product Backlog: Managing Work Efficiently

The Product Backlog is a list of items required to complete the product. It includes:

- New features & ideas
- Bug fixes & improvements
- Technical debt resolution

Who Manages the Product Backlog?

Product Team (Product Manager, Owners, and Scrum Team)

Influenced by Users, Customers, Designers, Developers, Marketing, Research Teams

Why is the Product Backlog Important?

- Provides a unified view of work.
- Ensures transparency for stakeholders.
- Helps teams prioritize tasks efficiently.

Types of Backlogs

Idea Backlog , Product Backlog , Development Backlog , Icebox

Product Backlog Grooming/ Replenishment Meeting

Why is Backlog Grooming Important?

Ensures the development team fully understands the upcoming work.

Helps refine backlog items by breaking down epics into user stories.

Identifies dependencies, blockers, and follow-ups before sprint planning.

Improves sprint efficiency by making sure only well-defined user stories enter the sprint.

Prevents low backlog situations where there aren't enough user stories for the team.

Key Participants in the Meeting

Product Manager (PM) , Developers, UX/UI Designers (if needed) ,Data/Analytics Team (if needed) ,Scrum Master (if applicable)

What Happens in a Backlog Grooming Meeting?

PM gathers customer/user/stakeholder insights and converts them into user stories.

Review the backlog to remove outdated or low-priority items.

Identify dependencies (e.g., backend work needed before frontend tasks).

Set an agenda (e.g., focus on epics, new features, tech debt, etc.).

Product Backlog Grooming/ Replenishment Meeting

During the Meeting (Execution)

Context Setting – PM explains the goal of each epic and user story.

Clarify User Stories – Answer questions and refine acceptance criteria.

Break Down Large Epics – Ensure stories are small, manageable, and testable.

Prioritize Stories – Order backlog based on value, urgency, and dependencies.

Estimate Effort – Use story points, T-shirt sizing, or planning poker for sizing.

Identify Risks & Challenges – Discuss technical feasibility and potential roadblocks.

Mark Follow-Ups – Note unanswered questions for further discussion.

After the Meeting (Follow-Up Actions)

Refine user stories based on discussion.

Update backlog to reflect changes.

Communicate decisions to stakeholders.

Prepare stories for sprint planning so development can start smoothly.

Challenges & How to Handle Them

Challenges & How to Handle Them

🚧 Challenge: Repeating discussions across multiple meetings.
🔹 Solution: Document decisions in a shared space (e.g., Confluence, Notion, Jira).

🚧 Challenge: Unclear or vague user stories.
🔹 Solution: Use the INVEST principle (Independent, Negotiable, Valuable, Estimable, Small, Testable).

🚧 Challenge: Technical debt or missing details in user stories.
🔹Solution: Identify gaps early and assign follow-ups to the right people.

🚧 Challenge: Overloaded backlog with too many tasks.
🔹 Solution: Regularly clean up and remove outdated/low-priority items.

Key Takeaways

Well-groomed backlogs improve sprint efficiency and reduce last-minute confusion.

PMs must prepare insights before the meeting to ensure valuable discussions.

User stories should be clear, small, and testable with defined acceptance criteria.

Technical feasibility, dependencies, and priorities must be discussed.

Product Backlog Grooming/ Replenishment Meeting

During the Meeting (Execution)

Context Setting – PM explains the goal of each epic and user story.

Clarify User Stories – Answer questions and refine acceptance criteria.

Break Down Large Epics – Ensure stories are small, manageable, and testable.

Prioritize Stories – Order backlog based on value, urgency, and dependencies.

Estimate Effort – Use story points, T-shirt sizing, or planning poker for sizing.

Identify Risks & Challenges – Discuss technical feasibility and potential roadblocks.

Mark Follow-Ups – Note unanswered questions for further discussion.

After the Meeting (Follow-Up Actions)

Refine user stories based on discussion.

Update backlog to reflect changes.

Communicate decisions to stakeholders.

Prepare stories for sprint planning so development can start smoothly.

Daily Standup Meeting (Scrum Standup)

Daily Standup Meeting (Scrum Standup)

What is a Standup?
A brief, focused session where team members discuss progress, challenges, and next steps. It ensures alignment and quick issue resolution.

Purpose of Standup Meetings:
Ensure project progress is on track,
Identify blockers early,
Foster team collaboration

Types of Standups: Round Robin Style, Walk the Board Style:

Round Robin Style:

A standup format where each team member individually shares their progress, covering three key points:

What did I do yesterday?

What will I do today?

Any blockers/issues?

Advantage: Everyone gets a chance to speak.

When to Use:

If the team follows Scrum methodology and works in sprints.

Good for structured, small teams where individual updates are important.

Daily Standup Meeting (Scrum Standup)

Walk the Board Style

A standup format where the team reviews the task board (Kanban/Sprint board) column by column, discussing the progress of each task rather than focusing on individual team members.

How It Works:

The team moves through each section (To Do → In Progress → Review → Done).

Any blockers or delays are addressed as they appear on the board.

Focus is on how the project is moving forward, not just individual updates.

When to Use:

If the team wants to focus on project flow rather than individual work.

Works best for Kanban teams or teams with continuous delivery.

Testing

Testing - Definition

Testing is the process of evaluating a product, feature, or system to identify errors, ensure functionality, and improve performance before releasing it to users. It helps ensure that the product meets user expectations and business goals while minimizing risks.

A/B Testing for Product Managers

Definition: A/B Testing (or Split Testing) is an experiment where two or more versions of a product or feature are compared to determine which performs better based on key metrics. How It Works:

Current State of the Product → Baseline performance before making any changes.

State After the Release → Changes or improvements introduced.

Measure the Impact → Analyze user behavior and key performance indicators (KPIs) to determine if the change was effective.

Key Parts of A/B Testing

Goal & Motivation → What are you testing and why?

Hypothesis → What do you expect to improve/change?

Test Groups:

Control Group (A): Users who continue using the current version.

Variation Group (B): Users who see the new version being tested.

Key Elements of A/B Testing

Key Elements of A/B Testing

Randomization → Users are randomly assigned to groups (ensures fairness).

Traffic Split → How users are divided (e.g., 50% Control, 50% Variation).

Duration → How long the test runs (e.g., 2 weeks, 1 month).

Variants → Different versions being tested (e.g., new button color, UI change).

Platform & Devices → Which platforms are included (Web, Mobile, iOS, Android).

Statistical Significance → Ensuring results are meaningful (not due to randomness).

Metrics Tracked → Conversion rate, bounce rate, click-through rate (CTR), etc.

How to Set Up an A/B Test

Test Proposal → Define the experiment, goals, and success criteria.

What to Improve → Identify the specific change (UI, feature, content, etc.).

How to Quantify the Impact → Choose relevant KPIs (e.g., increase signups by 5%).

Stakeholder Buy-in → Get approval from teams (PMs, designers, developers).

Track Progress & Analyze Data → Monitor results, measure impact, and decide whether to roll out or revert changes.

Why A/B Testing is Important for Product Managers?

Why A/B Testing is Important for Product Managers?

Helps make data-driven decisions instead of relying on assumptions.

Reduces risk of failure before launching a major feature.

Improves user experience and conversion rates over time

Aligns with business goals by optimizing features based on real user behavior.

Example Scenario:

Imagine you're a product manager for an e-commerce platform, and you want to increase the checkout completion rate. You hypothesize that changing the "Buy Now" button color from blue to green will make it more noticeable and encourage more purchases.

- Control Group (A): Sees the existing blue "Buy Now" button.
- Variation Group (B): Sees the new green "Buy Now" button.
- Metrics Tracked: Click-through rate, conversion rate, drop-off rate.

After running the test for two weeks, you find that the green button led to a 15% increase in purchases. Based on statistically significant results, you roll out the new design to all users, improving the platform’s revenue.
This process ensures data-backed decision-making, minimizing guesswork and maximizing impact.

Beta Testing: Definition & Key Aspects

What is Beta Testing?

Beta testing is the final phase of product testing where a nearly completed product is released to a limited group of real users in a real-world environment before the official launch. The goal is to identify issues, gather feedback, and improve product quality.

Goal of Beta Testing

Collect user feedback on product performance in real-world conditions

Identify bugs, usability issues, and edge cases

Ensure product stability, reliability, and user satisfaction

Validate whether the product meets business and user needs

Types of Beta Testing

Public (Open) Beta: Available to a large group of users, often through sign-ups or app stores (e.g., early access versions).

Closed (Private) Beta: Limited to a selected group (e.g., internal employees, early adopters, VIP users) under NDA.

Beta Testing: Definition & Key Aspects

Beta Testing Process

Establish Clear Goals – Define testing objectives (performance, usability, feature validation, etc.).

Identify Participants – Select testers that represent the target audience.

Create Documentation – Provide test guidelines, issue reporting templates, and FAQs.

Feedback Collection Process – Implement structured methods like surveys, user interviews, and bug reports.

Track Key Metrics – Measure beta performance using:

- Total number of beta users, Active beta users
- Total feedback collected, Actionable feedback
- Feedback implemented

Analyze Feedback & Implement Changes – Prioritize and resolve key issues before launch.

Advantages of Beta Testing

Uncovers Edge Cases – Finds unexpected issues that may not appear in internal testing.

Improves Product Quality – Helps refine UX, performance, and reliability.

Enhances User Trust – Engages users early, increasing adoption and retention.

Gives More Control Over Product Launch – Avoids major post-launch failures.

Beta Testing: Definition & Key Aspects

Beta Testing Process

Establish Clear Goals – Define testing objectives (performance, usability, feature validation, etc.).

Identify Participants – Select testers that represent the target audience.

Create Documentation – Provide test guidelines, issue reporting templates, and FAQs.

Feedback Collection Process – Implement structured methods like surveys, user interviews, and bug reports.

Track Key Metrics – Measure beta performance using:

- Total number of beta users, Active beta users
- Total feedback collected, Actionable feedback , Feedback implemented

Analyze Feedback & Implement Changes – Prioritize and resolve key issues before launch.

Advantages of Beta Testing

Uncovers Edge Cases – Finds unexpected issues that may not appear in internal testing.

Improves Product Quality – Helps refine UX, performance, and reliability.

Enhances User Trust – Engages users early, increasing adoption and retention.

Gives More Control Over Product Launch – Avoids major post-launch failures.

Data-Driven Product Management

Understanding Data-Driven Product Management

After launching or releasing a product, it is essential to measure its success through data. Data-driven product management helps organizations make informed decisions by following a structured approach:

Data-Driven Product Management

The 5-Step Framework for Data-Driven Product Management

Start with the Right Questions

What do we need to measure and why?

What data do we want to track?

How does this data drive decision-making?

Who will use this data, and how will they access it?

Define Key Product Metrics

Metrics should be easy to understand and provide meaningful insights.

Product teams should align these metrics with business goals.

Identify the Right Data Sources

Determine where the data is collected and how it is tracked.

Ensure data accuracy and consistency to avoid redundancy over time.

Communicate Product Metrics Effectively

Clearly explain metrics, their impact, and how they align with strategy.

Define abbreviations, provide context, and automate reporting.

Automate Data Processes

Streamline data collection and reporting to reduce manual efforts.

Key Product Metrics

User Engagement & Retention Metrics

Churn Rate → Percentage of users who stop using the product over time.

Retention Rate → Percentage of users who continue using the product.

Customer Lifetime Value (CLV) → The total value a customer brings during their entire engagement.

Net Promoter Score (NPS) → Measures customer loyalty and satisfaction (1-10 scale).

Customer Satisfaction Score (CSAT) → Measures user satisfaction on a 1-5 scale.

Customer Effort Score (CES) → Measures ease of interaction with the product.

User Behavior Metrics

Daily Active Users (DAU) → Users interacting with the product daily.

Monthly Active Users (MAU) → Users interacting with the product monthly.

Product Stickiness (DAU/MAU Ratio) → Indicates how often users return.

Feature Adoption Rate → Percentage of users adopting a new feature.

Average Session Duration → Measures time spent in meaningful interactions.

Lead & Sales Metrics

Marketing Qualified Leads (MQLs) → Leads showing interest due to marketing efforts.

Sales Qualified Leads (SQLs) → Leads ready to talk to the sales team.

Product Qualified Leads (PQLs) → Users who have experienced value and are likely to convert.

Key Product Metrics

Marketing & Acquisition Metrics

Bounce Rate → Percentage of users who leave after visiting only one page.

Customer Acquisition Cost (CAC) → Total cost spent to acquire a customer.

Click-Through Rate (CTR) → Percentage of users clicking on a specific element (e.g., ad, button).

Organic Traffic → Number of users coming from unpaid sources (SEO, referrals).

Paid Traffic → Users acquired through paid campaigns.

Number of App Installs → Measures mobile app adoption.

App Store Rating → Measures app performance based on user reviews.

Revenue & Business Metrics

Monthly Recurring Revenue (MRR) → Predictable revenue earned monthly.

Average Revenue Per Account (ARPA) → Revenue generated from each customer account.

Return on Investment (ROI) → Measures profitability of investments in the product.

Virality (K-Factor) → How fast a product spreads through referrals.

Time to Value (TTV) → Time it takes for a user to realize the product's value.

Product Launch vs. Product Release

Product Launch vs. Product Release

Product Launch and Product Release are two different things, often misunderstood. Here’s a clear breakdown:

Product Release

- A product release happens when a new feature, update, or improvement is made available to users.
- Releases happen frequently (sometimes weekly or even daily).
- It is primarily a technical event, handled by engineering and product teams.
- No marketing campaign is needed—just a simple, clear explanation for users.

Key Components of a Product Release:

What has changed?, How will it benefit users?, Who will be affected?, Will it require user adaptation or training?

Product Launch

A product launch is a major event that introduces a new product (or a significant update) to the market.

It involves multiple teams: product, sales, marketing, finance, support, and customer success.

A launch requires a strategic marketing plan, including:

- Press releases , Advertising campaigns ,Sales training , Customer education

API for Product Managers (PMs)

What is an API?

API (Application Programming Interface) is a set of rules that allows different software applications to communicate with each other.

Example: Just like a waiter takes your order to the kitchen and brings back food, an API takes user requests, fetches data, and returns results.

Why Should Product Managers Care About APIs?

Helps in defining product features that rely on external services

Ensures smooth collaboration with engineering teams

Supports integration decisions (whether to build or buy an API)

Understands API limitations & impact on performance

Key API Concepts for PMs

Request & Response – APIs take input and return data.

Endpoints – Specific URLs where API requests are sent (e.g., /users/profile).

Authentication – Uses keys or tokens to verify access (e.g., OAuth, API keys).

Rate Limits – Restrictions on API usage to prevent overload.

Performance & Latency – Speed impacts user experience.

REST vs. GraphQL – REST uses multiple endpoints; GraphQL allows flexible data fetching.

API for Product Managers (PMs)

What is an API?

API (Application Programming Interface) is a set of rules that allows different software applications to communicate with each other.

Example: Just like a waiter takes your order to the kitchen and brings back food, an API takes user requests, fetches data, and returns results.

Why Should Product Managers Care About APIs?

Helps in defining product features that rely on external services

Ensures smooth collaboration with engineering teams

Supports integration decisions (whether to build or buy an API)

Understands API limitations & impact on performance

Key API Concepts for PMs

Request & Response – APIs take input and return data.

Endpoints – Specific URLs where API requests are sent (e.g., /users/profile).

Authentication – Uses keys or tokens to verify access (e.g., OAuth, API keys).

Rate Limits – Restrictions on API usage to prevent overload.

Performance & Latency – Speed impacts user experience.

REST vs. GraphQL – REST uses multiple endpoints; GraphQL allows flexible data fetching.

Thank You

We hope you found these notes useful for learning Product Management

📧 varadkoppar3@gmail.com

Feel free to reach out for feedback, collaboration, or any questions!