

FROM IDEA TO ADOPTION: PRODUCT THINKING FOR AVALANCHE BUILDERS

KEY TOPICS

- Frame Problems
- Define Target Users
- Understand Market Fit & Key Competitors
- Define Product Unique Value Proposition
- Create User-Centered MVPs

framing problems

“Every problem has a solution ...
But all solutions do not solve problems.”

Solution-Driven Thinking

Why You Should NOT Start With the Solution

When you start with the solution, your brain does something dangerous:

- You begin searching for a problem that **justifies your idea**
- This leads to **artificial use cases**
- **Unnecessary** blockchain usage
- Features **nobody asked for**
- Products that **only builders understand**



"I've come up with a brilliant solution. Your job is to find a problem that fits it."

Problem Framing

The 4 parts of a Problem Framing process

A good, complete problem framing process will tell you 4 things:

- **What is the problem?** What are the struggles? What task needs to be accomplished? What pain point needs to be relieved?
- **Who is affected?** Who is experiencing the problem? Can this user be further specified (by persona, motivation, reason for being in the situation)?
- **Where does it happen?** What is the context in which the user experiences the problem? Is it in a physical or digital space? Who else is involved?
- **Why is this problem worth solving?** What value does it bring to the user? Why is blockchain the appropriate technology to solve this problem?



Problem Statement

Defining your Problem Statement

A rigorous user-framing process enables you to define a focused and well-articulated problem statement — one that will guide you through the following research, ideation, product development phases and pitching.

The articulated Problem Statements should be:

- **Brief** — If you can condense your problem statement down to a few sentences, others will quickly understand what you focus on and why, and what's out of scope.
- **Factual & Unbiased** — Look at all of the information you have with an open, honest mind
- **Rooted** — When we jump to conclusions we often solve the symptoms, while real problem statements focus on the root cause
- **Positive & Negatives** — Problem statements can also capture opportunities, not only problems.



Bitcoin Example

■ Who is Affected?

Target Users

- Internet users exchanging value
- Online merchants
- Payers & payees who don't trust each other

Motivations

- Direct transactions
- Lower fees
- Final settlement
- Transfer digital value safely without a trusted third party

■ What is the Problem?

Core Issues

- Trust dependency in digital money systems
- Digital payments require trusted intermediaries
- Double-spending risk
- High fees and delays
- Transaction reversals
- Reliance on centralized trust

■ Where Does it Happen

Context

- Digital environments
- Online commerce
- Internet payments

Characteristics

- Global, remote transactions
- No shared jurisdiction
- No prior trust

Traditionally Involves

- Banks
- Payment processors
- Clearing houses

■ Why Does It Matter?

Market Context

- Online commerce is growing
- Trust-based systems scale poorly
- Intermediaries add cost, risk, and friction

User Value

- Trustless transactions
- Lower fees / Final settlement
- Verifiable ownership

System Value

- Decentralized resilience
- Self-sustaining network
- Security via incentives

WHO



WHAT



WHERE



WHY



Bitcoin Example

One-Sentence Statement

“Bitcoin solves the problem of trust-dependent digital payments by enabling peer-to-peer electronic cash through cryptographic proof, eliminating the need for centralized intermediaries while preventing frauds and double-spending in a low-cost, resilient, decentralized network.”

KEY TAKEAWAY

“Problem framing is not merely about identifying a problem...

but about understanding its context, implications, value, viability and potential for innovative solutions”.

defining user segments and mental models

“We can’t create a Digital Product for everyone.”

Define Target User

Why it's important?

1. It turns vague problems into solvable ones

“Users” is not a usable product input. Clear personas let teams frame problems precisely instead of building generic features based on assumptions.

4. It improves product–market fit

Strong products aim to be exceptionally valuable to a specific group. This depth is what leads to real adoption and retention.

2. It aligns the entire team around the same user

Personas create a shared language so decisions are driven by the needs of a clearly defined user, guiding product roadmap and backlog direction.

5. Removes bias preventing solution-first thinking

Without personas, teams often jump straight to features or technology. Personas force a user-first approach, leaving feature or functional decisions for last.

3. It enables better prioritization & user-centric approach

Every product team faces trade-offs. When a persona is defined, prioritization becomes strategic instead of political, placing users priorities first.

6. It sharpens design and UX decisions

Design is about trade-offs: simplicity vs. power, guidance vs. flexibility. This leads to more intuitive, intentional experiences.

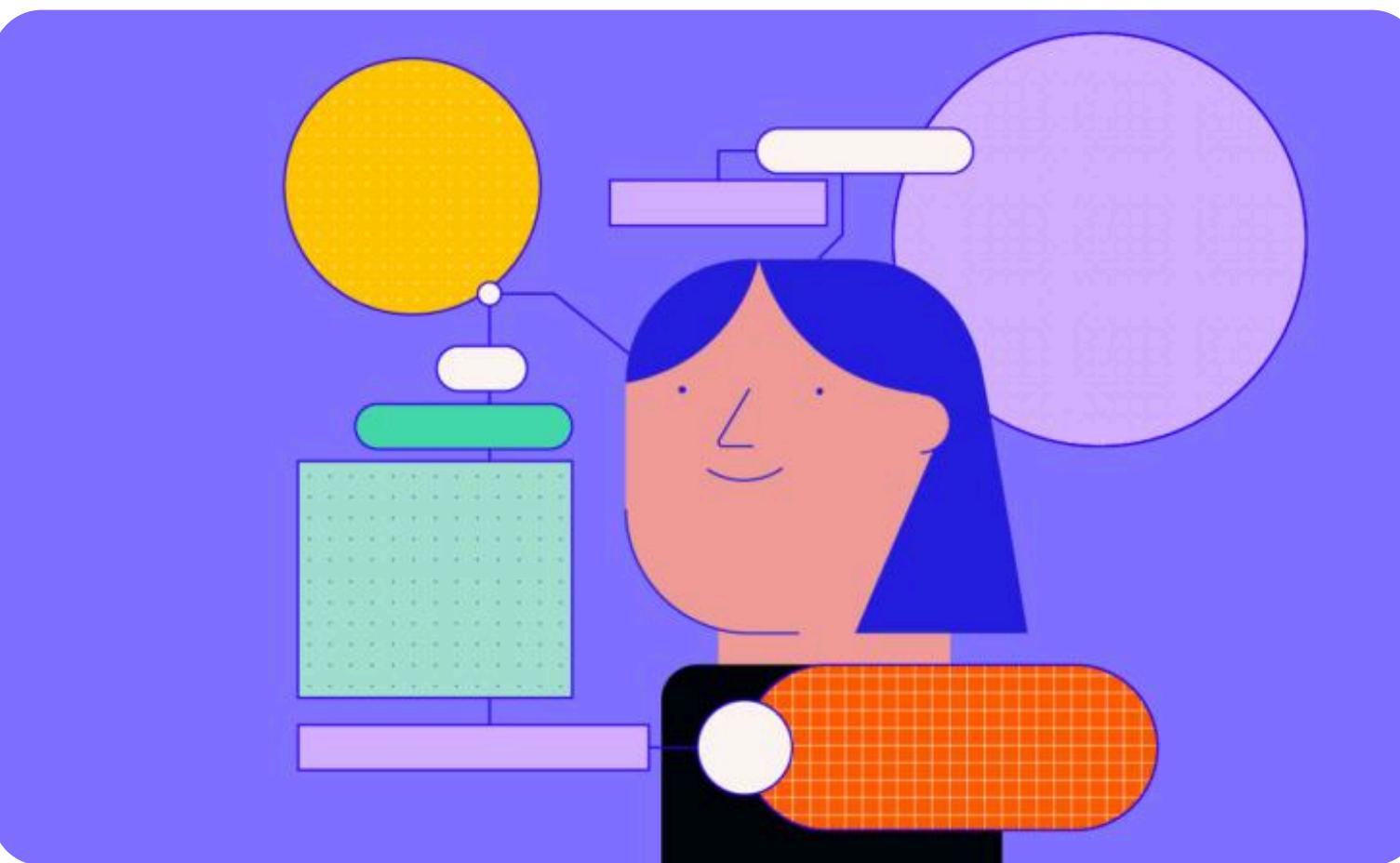
Personas, Segments & Mental Models

Defining Segments, User Personas & Mental Models

There are 3 interdependent aspects you need to explore in order to define your target user.

- **Segments** define who you are building for
- **Mental models** define how they think
- **Personas** make it usable for product decisions

- **Without personas**, segments stay abstract.
- **Without segments**, personas become fictional.
- **Without mental models**, UX becomes misaligned.



Personas, Segments & Mental Models

Defining Segments, User Personas & Mental Models

How to elaborate this user analysis into meaningful information and insights:

1. Define the Segment across 4 dimensions:

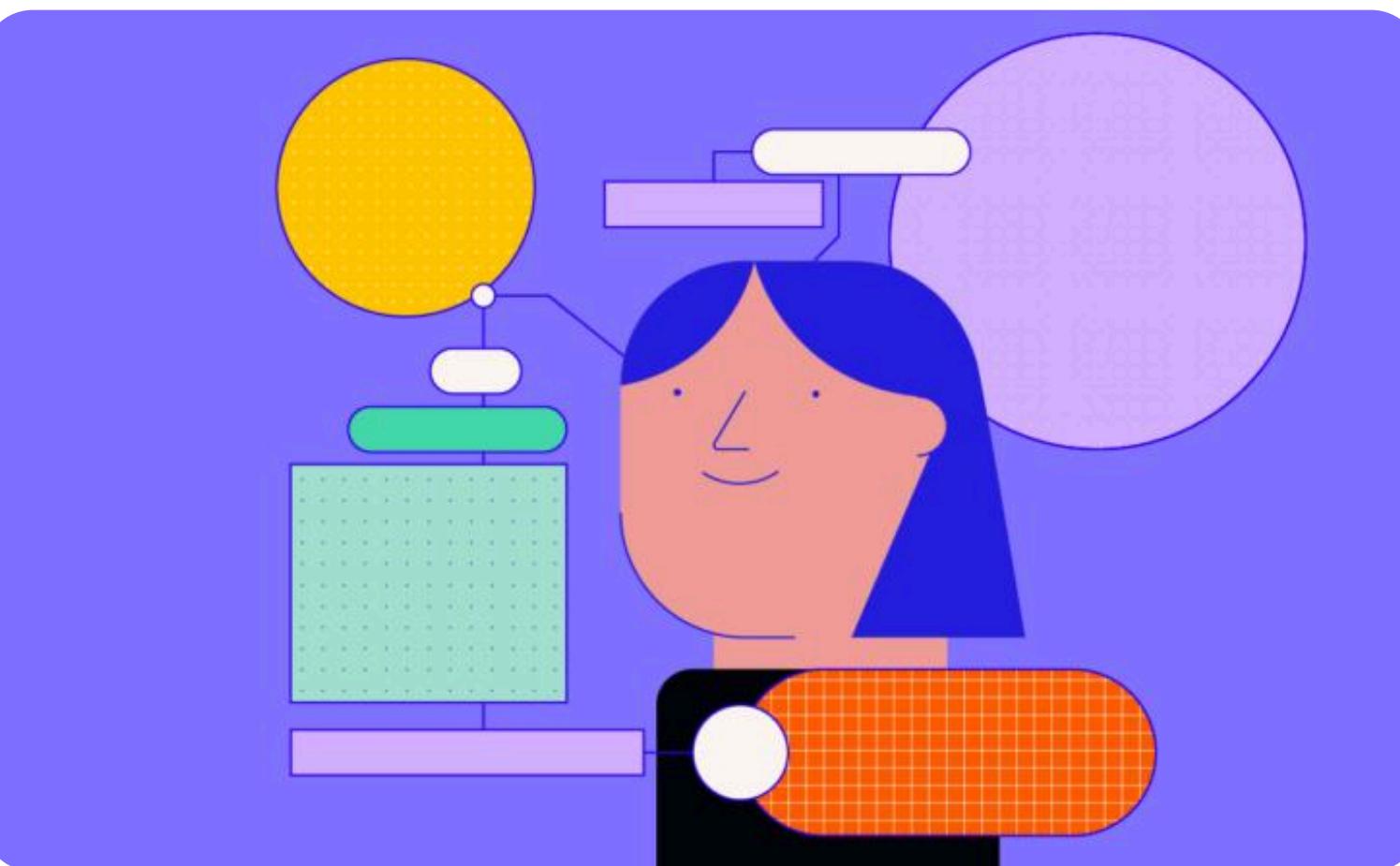
- Goals – What are they trying to achieve?
- Fears – What are they afraid of?
- Knowledge – What do they already understand?
- Behavior – How do they currently solve this?

2. Use Mental Models to explain:

- What does this user believe about how the system works?

3. Create a User Persona to define:

- A realistic, specific representation of your primary user inside a segment.



Bitcoin Example

■ Facts

- Uses traditional banking apps daily
- Makes frequent transactions (peer-to-peer, rent, bills, subscriptions)
- Mid-level tech literacy (understands apps, not infrastructure)
- Occasionally invests using mainstream fintech platforms
- Values security and reliability over experimentation

■ Pain Points

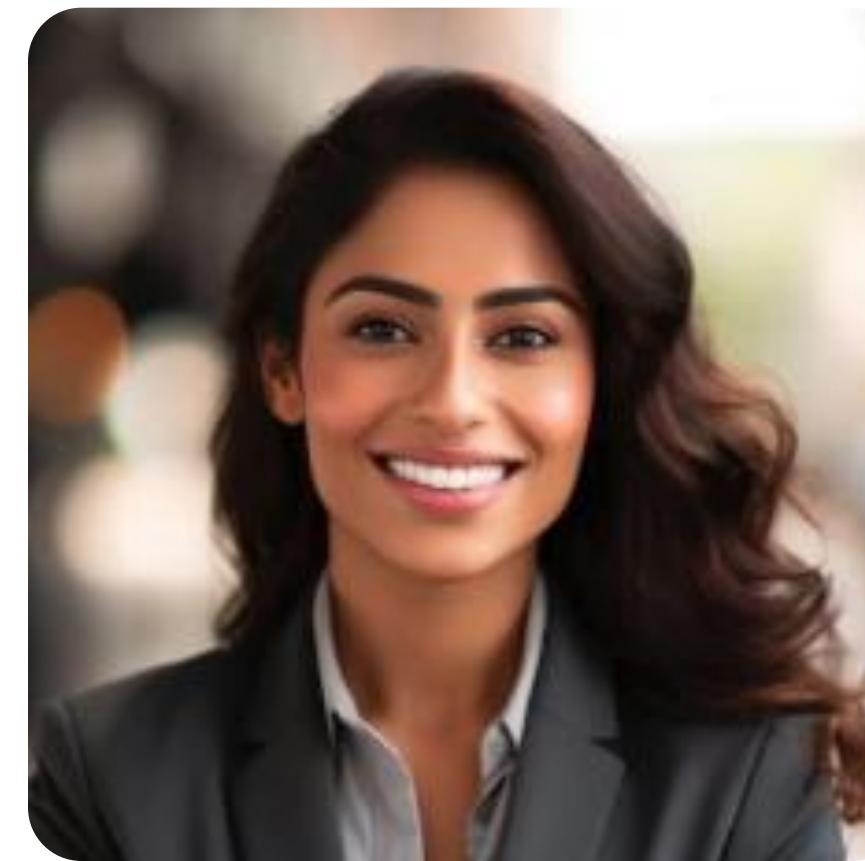
- International transfers are slow / expensive
- Hidden fees in cross-border payments
- Confusing transaction processing times
- Reversals and pending payments create uncertainty
- Fear of fraud or sending money to the wrong person
- Limited transparency on where money is “in transit”
- Has to rely completely on bank policies and customer support

■ Tasks & Behaviors

- Checks bank app multiple times per week
- Sends money to friends and family regularly
- Pays & subscriptions bills digitally
- Expects instant confirmation after sending money
- Reads transaction summaries but skips detailed terms
- Trusts brand reputation more than technical explanations
- Avoids complex financial tools

■ Goals

- Send money quickly and confidently
- Minimize fees
- Get immediate and final confirmation
- Feel secure while transacting
- Avoid technical complexity
- Have clear transaction history and tracking
- Resolve issues quickly if something goes wrong
- Maintain control over finances without needing expert knowledge



Jane, 34 years

Personas, Segments & Mental Models

How would these aspects influence my product?

Your User Persona characteristics will determine:

1. MVP Feature Scope & Prioritization
2. Complexity Level
3. Trust & Safety Design
4. Language & Vocabulary
5. Overall User Experience
6. Product Positioning & Messaging
7. Growth Strategy



Jane, 34 years

KEY TAKEAWAY

“Every product decision is a user decision in
disguise...

If you don’t define the user, the product defines
itself or by assumptions — and usually becomes
too complex”.

competitors research & market fit

“Your real competition is anything that already solves the user’s problem.”

Market Competitors

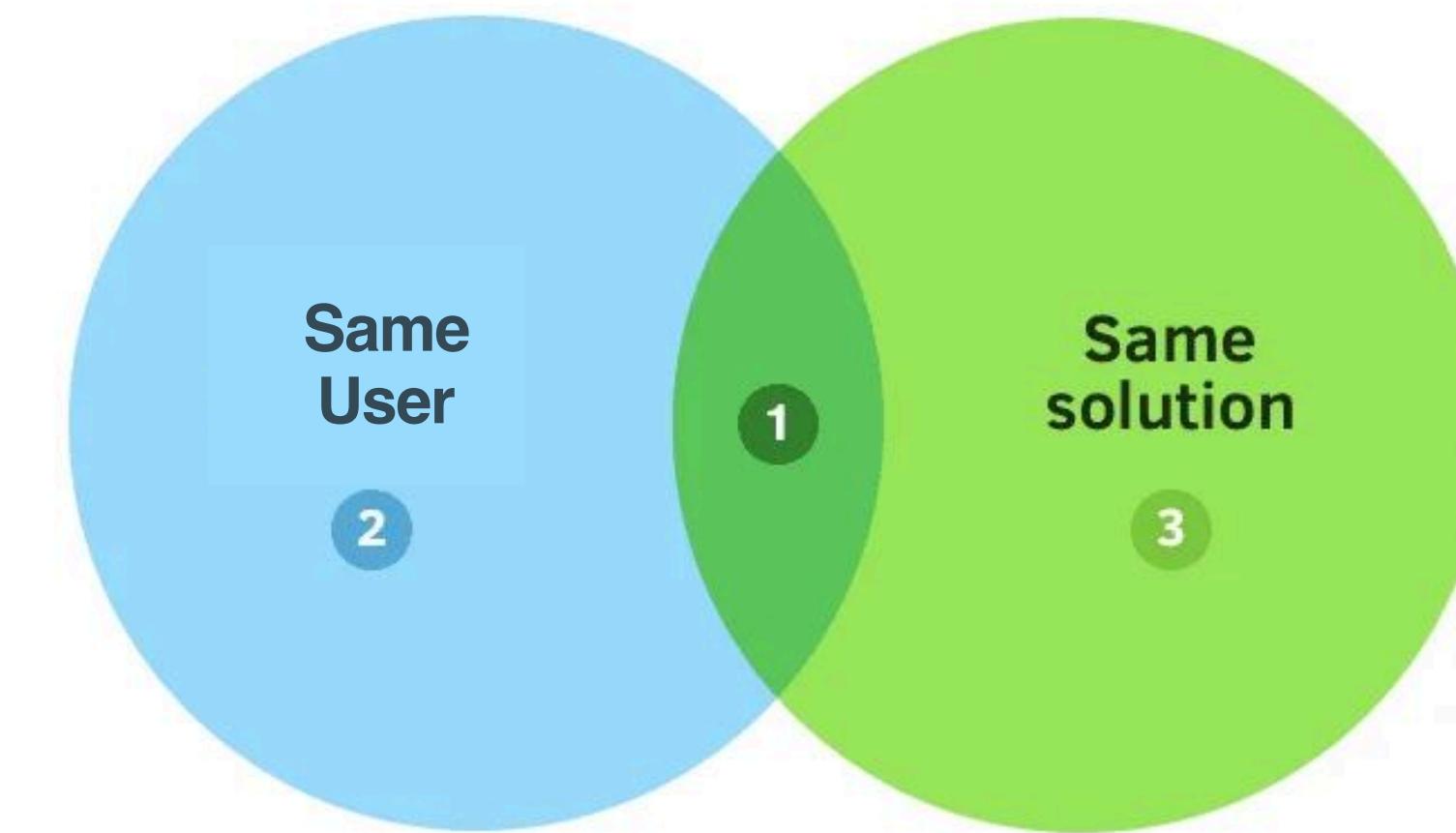
It's not just about features and technology

Potential users compare you against products that address the same need as your MVP, including:

- Existing Web3 tools (Inside/Outside Avalanche)
- Other Web2 tools
- Their current behavior & use patterns

You need to understand your competitors in order to:

- Identify your idea's weakness, strengths, opportunities
- Learn what worked or failed in the past
- Find your unique value proposition



① Direct competitors

Solving the same problem for the same customer with a similar product.

② Different solution

Solving the same problem for the same customer, but in a different way.

③ Different customer

Solving the same problem in a similar way, but for customers in a different industry/vertical.

Market Fit

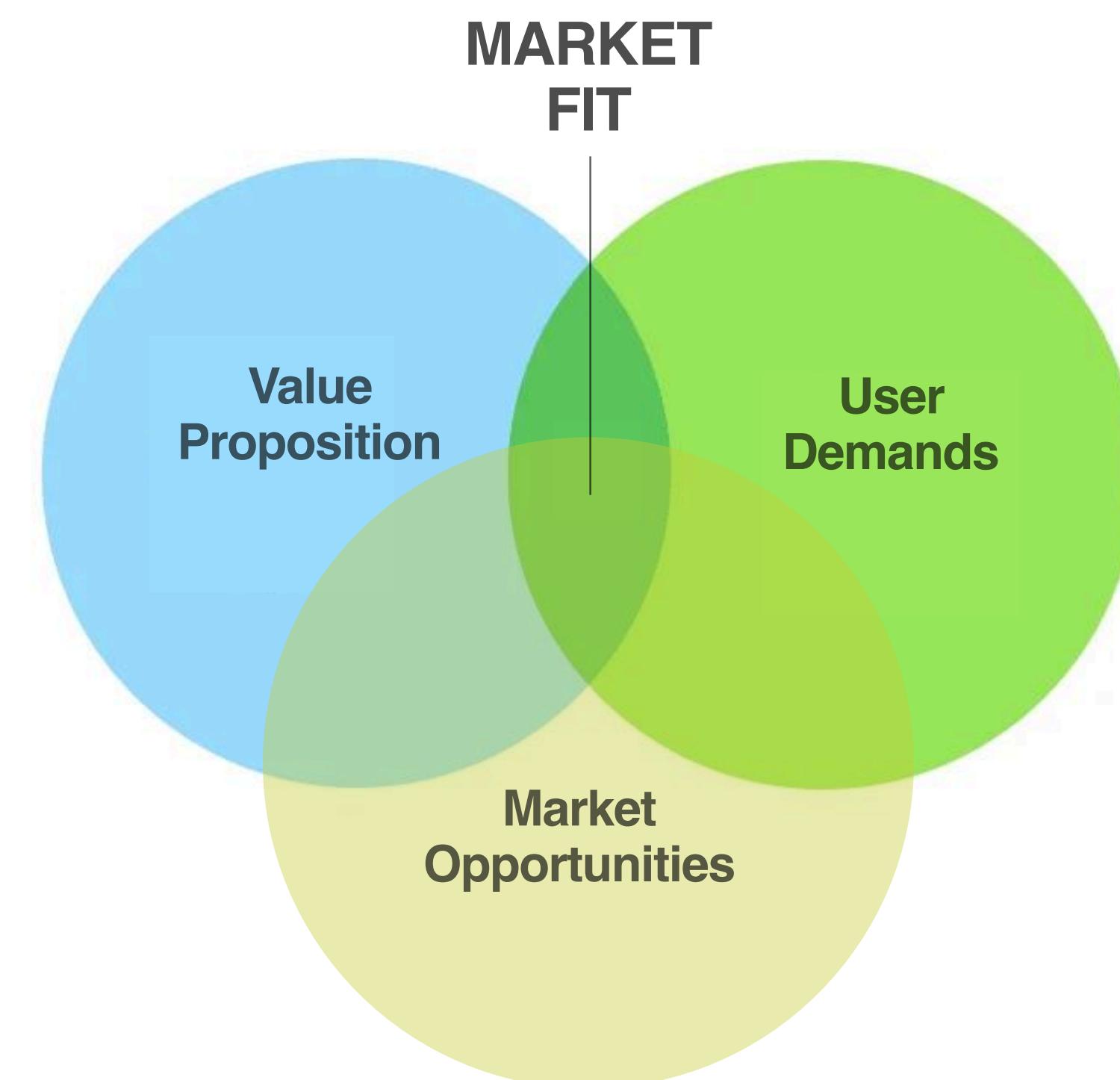
Market fit

Many projects are technically impressive, but:

- Solve edge-case problems
- Serve crypto-native users only
- Compete with 10 identical Web3 apps
- Depend on various incentives to attract users

Market fit thinking forces teams to ask:

- Who has this problem today?
- Is this a top-3 pain in their life?
- How are they solving it now?
- Why would they switch to my solution?



KEY TAKEAWAY

“You’re not just competing with other similar platforms...

You’re also competing with current habits and solutions to similar needs”.

product direction & value proposition

“If the first thing you explain is the tech, you’re building features...

If the first thing you explain is the user problem, you’re building a product.”

Product Strategy

Core Purpose, Product Direction & Value Proposition

Technology is not the product, you need to define your product strategy.

- **Core purpose** answers why it matters.
 - Why does this product deserve to exist?
- **Product direction** is your strategic focus.
 - A specific user
 - A specific use case
 - A specific outcome
- **Value proposition** is the reason someone would choose you.
 - Why is this better than what they use today?
 - Why is this better than other similar Web3 solutions?
 - What is my unique or differential value proposition?



Product Strategy Example - Bitcoin

Core purpose

- It replaces institutional trust with cryptographic proof so two parties can transact directly over the internet without needing banks or payment processors to prevent fraud or double-spending.
- It solves how to achieve distributed consensus on transaction history without central authority

Product direction

A specific user:

Internet participants who:

- Want to send or receive digital payments,
- Do not trust each other,
- Want to avoid reliance on centralized financial institutions

A specific use case:

- Remote transactions
- Cross-border payments
- Situations where trust is low

A specific outcome:

- Transactions are irreversible after confirmation;
- Double-spending is computationally impractical;
- Settlement is objective and verifiable by anyone;
- No third party is required.

Value proposition

1. Why is this better than what they use today?

- No intermediary required
- Lower Costs
- Speed & Process efficiency
- Cryptographic verification instead of institutional trust

2. What is the differential value proposition?

- First solution to decentralized double-spend prevention
- Trustless transaction verification
- Tamper Proof
- Permissionless participation
- Protocol-level neutrality and censorship resistance

KEY TAKEAWAY

“Clarity of purpose drives focus. Focus creates differentiation...Differentiation creates adoption”.

user-centric approach: from needs to product decisions

“User-centric means every product decision starts from user reality — not from technology capability.”

User-Centric Products

We're always building products for someone else

A common mistake is using ourselves as the primary reference for product decisions.

As Web3 enthusiasts we tend to assume:

- "All users care about decentralization."
- "All users understand wallets."
- "All users want to manage private keys."

As a Result, most Web3 products assume users are:

- Technically literate and familiar with wallets
- Comfortable with risk
- Fluent in crypto language

The truth is most users care about convenience:

- Solving their problem
- Feeling safe & not losing money
- Having a small learning curve
- Not getting overwhelmed



User-Centric Products

What “User-Centric” Actually Means in Web3

There are a few key principles for building user-centric products:

- Prioritize solving real user needs — not just a blockchain use case
- Abstract unnecessary crypto mechanics
- Match the user’s mental model
- Assume zero prior knowledge & minimizes cognitive load
- Design for irreversible actions — prevents mistakes before they happen
- Provide error recovery guidance and safety education
- Build trust through transparency
- Warn about irreversible actions & prevents dangerous actions
- Never leaves users waiting without feedback



#1 Abstract Complexity

What “User-Centric” Actually Means in Web3

Users shouldn't need to know technical terminology, processes, or concepts to use a Web3 product. You need to:

- Automatize/Hide as many steps as possible
- Remove unnecessary decisions
- Hide technical language
- Prioritize user habits & learned behaviors
- Avoid overwhelming technical interfaces

UX Best Practices

- Account Abstraction
- Biometric logins
- Social Recovery
- Seedless wallets
- Meta TXs.
- Human-Readable Tx Signing.

#2 Design for Familiarity

Build Web3 products that feel recognizable

A user-centric Web3 product minimizes conceptual & behavioral change:

- Maps blockchain actions to real-world concepts (pay, own, transfer, confirm)
- Preserves established digital behaviors (bank transfers, checkout flows, login patterns)
- Introduces new Web3 concepts progressively, not all at once
- Aligns with existing mental models from Web2 and crypto-native tools
- Innovate in technology & infrastructure — not in basic behavior

UX Best Practices

- Use familiar confirmation patterns
- Provide clear transaction states
- Use safe defaults to reduce decision fatigue
- Highlight irreversible actions visually
- Keep terminology consistent and familiar

#3 Design for Trust

What “User-Centric” Actually Means in Web3

To adopt a product, users need clear signals of trust and verifiable reputation:

- Clear risk warnings & irreversible actions
- Makes costs predictable & Simple (no surprise fees)
- Error recovery & prevention patterns
- Transaction simulation features
- Human-readable contract actions
- Scam protection & Security Audits Validation
- Show status & confirms success in simple language

UX Best Practices

- Audit Badges
- Last Verified Stamp
- Contract Verification
- On-chain Reputation
- Error Prevention & Recovery
- Progressive information disclosure

#4 Validate with Users

Assumptions create features — validation creates products

If you don't validate with real users, you're not building a product...you're building a belief:

- Every feature and backlog is based on a hypothesis
- Every UX decision is an assumption about user behavior
- Every product idea assumes a real need exists
- Every token model assumes users care about incentives
- Every interface design assumes users understand our UX/UI flow

UX Best Practices

- User interviews
- Task-based usability testing
- Wait-list sign up
- A/B testings

KEY TAKEAWAY

“Usability is a key competitive advantage — not decentralization alone...

Adoption is driven by clarity, not just technology”.

Closing Thoughts

- These steps are interconnected and necessary for developing solid product ideas.
- They are not simple exercises — they prevent wasting time, effort, and resources at early product stages.
- They help avoid building ideas, features, or products without real impact or adoption.
- A structured approach enables small, controlled iterations instead of starting over if product fails. You can adjust components independently.
- This allows continuous improvement until the product achieves traction.

