Tips to Validate Your Startup Idea

techready women

Summary by

InhouseVentures

Overview

Before investing time and money into building a product, founders should focus on understanding the problem they're solving and validating it with real customers. In this lesson, Crystal McGregor, General Manager at Tech Ready Women Academy, shares practical, experience-driven advice for testing ideas, engaging early users, and generating revenue before writing a single line of code. Her tips emphasise customer insight, scrappy prototyping, and iterative learning over perfection, helping founders avoid costly missteps and build solutions that truly resonate.

20-Point Summary:

1. Start with the problem, not the product — focus on solving real human problems before building technology.
2. Avoid assuming your idea is viable just because friends say it is.
3. Conduct proper customer validation — interviews, focus groups, surveys — beyond your personal network.
4. Take customers on a journey of understanding, learning their pain points and current solutions.
5. Ensure you understand how the problem might be solved better in the future.
6. Use a "sticky tape solution" — piece together existing tools like Canva, Airtable, Notion, or Zapier to prototype.
7. Be prepared for manual "heavy lifting" in the early stages to connect tools and manage the experience.
8. Run the journey with existing tech to identify drop-off points and usability issues before investing in development.
9. Define your core initial user clearly — not "everyone," but a specific first target audience.
10. Understand demographics, needs, and motivations of that first customer segment deeply.

11. Recognise early adopters may differ from your long-term mainstream customers.
12. Focus on getting first customers to commit financially — even a small payment matters.
13. Monetisation in the MVP stage validates trust and value.
14. Paid feedback is more reliable than free feedback.
15. Avoid waiting for a perfect product before launching — perfection delays learning.
16. Embrace small, rapid experiments to test assumptions.
17. When building, build for learning, not for perfection.
18. Engage real users early to discover what they actually care about.
19. Iterate based on real-world usage and feedback.
20. Your ultimate aim is to evolve with your users, not to deliver a “finished” first version.