DMMT!

Trust the process.

Below is a 7-step, ideation-to-ship process I've developed over the last 8 years while creating amazing products with exceptional product, engineering, and marketing teams.

VIEW ONLINE

01 / 07

Why?

Why are we building this, how does it move the needle forward?

Define the business objectives

1. What problem are we solving, is this the highest priority?
2. Are all of the stakeholders aligned on the problem?
3. What will this change cost us, and how are we going to measure its KPIs and ROI?
4. Define your hypothesis, knowns vs unknowns

TIP: Customer data is the great equalizer when priority differences occur with stakeholders. Bring customer feedback, insights and analytics to the table to help back your why.

How?

How are we going to create the solution?

Define the product objectives.

1. What are the new features or existing feature changes that will solve the problem?
2. Collaborate with team leads to define the key workflows, customer data drive boxes and arrows that become high-fidelity screens we can use for prototype testing later.
3. What are the knowns vs the unknowns? Turn the unknowns into knowns by asking questions via feedback forms and user interviews.
4. Create a one-pager outlining the objective, rally the team leads, and create excitement!

TIP: Focus on the minimal amount of features/fixes that will drive the biggest impact. ROI is everything, especially for startups. Remember you can do fast-follow releases, and get the solution out ASAP.

Validate.

Validate your solution and remove ambiguity by testing your solution with users.

Before investing in the build phase, test and validate your solution.

1. Keep it simple, design some low-fi prototypes, or even a feedback form, take the feedback and modify your solution accordingly.
2. Your user testing forms or interview questions should involve the unknowns from your original hypothesis. Take these findings and update your solution.

TIP: Focus on the minimum required effort to validate your solution, depending on complexity, a simple 5-minute form will do. Incentive offers can make a HUGE difference when getting users to opt-in.

Design.

Take the business objectives, solutions, and feedback and start designing the high-fidelity feature.

Turn low-fi into hi-fi.

1. Define the MVP key workflows, screens and UX goals.
2. Validate any technical concerns, and constraints with engineering, and adjust the feature accordingly.
3. Work with the product manager and define the build sprints and roadmap. Review these sprints and designs/key workflows with engineering, operations and marketing teams– timelines and alignment are critical to a successful launch.
4. Get designing, and prioritizing the product roadmap using existing components and templates, outline any new ones and add them to your design system.
5. Use your hi-fidelity screens to create prototypes, demo these with engineering and product teams, get feedback and create a deeper understanding of what is about to be built.

TIP: Try to minimize net new components and logic when fixing issues, this will greatly extend development time and add a new layer of testing.

Test/QA

Internally test the feature, look for bugs and squash them.

Test it, try to break it.

1. Do an internal bug-bash. Test your pre-production feature with the team and look for edge cases, bugginess, modality, device and differing OS issues.
2. Prioritize your bug fixes, and what needs to be done vs. fast-follows. Stick to your release date.
3. Do a final QA check after bugs are fixed, and prep for release!
4. Align the stakeholders with the release plan, and get all collateral campaign materials ready.

TIP: Get everyone in a room/zoom and bug-bash together if possible, the real-time collaborative efforts can be time/cost efficient and result in quicker solutions.

Release.

Push to production, all hands on deck for user support, feedback and bugs!

Test it, try to break it.

1. Get your release plan in action, what are the post-release action items for the team, customer support, bug triage, feedback, and most importantly user engagement!
2. Prioritize your outstanding bug fixes, or secondary feature updates from the bug bash, and push them out.
3. Process any support or feedback issues, and address them in order of importance.
4. Start measuring KPIs, what's working, and what's not, and build on the learnings.

TIP: Make your post-release plan BEFORE THE RELEASE! Be efficient with time and expectations. Bugs erode user trust, while nothing is ever 100% bug-free, work as hard as you can to reduce the bugs that get to your users.

Retro.

Sure the release is a team celebration, but more importantly, what did we learn as a team?

Better, faster, stronger…

1. Rally the team, and do a post-release retrospective. Give everyone involved in the project a moment to speak about what they liked, and what they didn't. Self-reflection is very important here, how could you have done better?
2. Take your retro learnings and apply them to your next product release

TIP: Be flexible, and adapt, there is always something to improve. Create an inclusive, supportive environment where team members can share openly the good and bad. Don't cover the truth, but be respectful in how you present it.