

**Objective:**  
Deliver a Minimum Viable Product/ Service (MVP) that will delight and engage your users.

## Product Development Tools, Tips, Tactics, Checklists and Strategies

### Product Development- THREE MOST IMPORTANT THINGS...

- 1) Before you start live product/service development be sure you are really ready to start. See checklist.
- 2) Clearly define A, B, and C priority items based on user needs and expected revenue. Be ready to change these priorities as needed.
- 3) Avoid using untested resources including people and technology. Test before the project starts.

### GOALS, OBJECTIVES, & DELIVERABLES

- Determine if you are really ready to start development.
- Essential elements for your development plan.
- Plan B – What to do when/if things go wrong. A contingency plan.
- INITIAL DELIVERABLE: End-to-end executable development plan, with specifications, timeline, budget, and contingency plan.
- FINAL DELIVERABLE: A Minimum Viable Product (MVP) or service.

### HUMAN RESOURCES NEEDED TO BUILD AN MVP

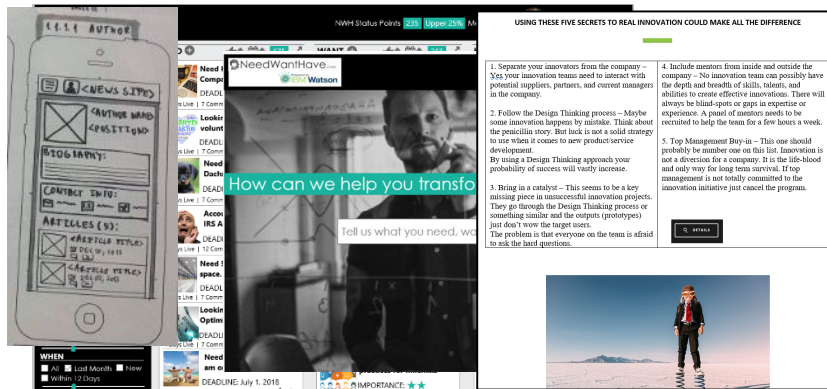
- An experienced project manager who is fully committed to the project. <- Real Ownership!
- An experienced lead programmer. An experienced tester.
- A graphic artist. Note: Don't let programmers do graphics work.
- A marketing person to represent target users and plan learning launch.

### HOW TO IDENTIFY AND ENGAGE HUMAN RESOURCES

- This is called recruitment... It's an ongoing effort.
- Keep a file with key contacts and constantly build that file. Schedule informal meetings. (Project managers, programmers, testers, artists, etc...)
- Referrals, networking, past projects, special interest groups, online resources like Upwork.com.
- Last choice... Recruiters. (But sometimes the only way.)
- Look for small projects to test people on your contacts list. Constantly test... (It's the only way to know for sure)

### Why is it important to really be ready to start development before you BEGIN?

1. Once you start development you will be spending money quickly.
2. Once you start development your team will expand quickly.
3. It gets very expensive to make design changes once active development begins. (Even if you're not "paying" for development.)
4. If your budget runs out before you complete development it will be a disaster for your company, your investors, your employees, co-founders, and you.
5. WARNING: Do not flip the "Development Switch" on until you are really ready to start. The downside is huge.



### ARE YOU READY TO START PRODUCT DEVELOPMENT – A CHECKLIST

- You know what you will make (at least 90% sure). This is about the making not about the dreaming, talking, or designing.
- You have a Project Manager who will own the project. That person must be involved and lead the entire project from end-to-end.
- Solid idea tested with target users. Your main assumptions for behavior change are proven. How have you proved this out?
- Tight, tested, prototypes. (Even if they are just on paper) – Every page, option, menu item. You need to decide not someone else.
- The product/service specification – In one document. Precisely define the product. Use pictures. If more than 10% of your spec is still not really known. Then work on your prototypes until they are known. This is not the time to experiment.
- Keep it simple at first – Go for the Minimum Viable Product (MVP) but have your other feature ideas prioritize just in case you have time/budget. Or for you developers to work on while you are running tests.
- You have a clear product feature list with every element tagged as A, B, or C priority. A= Must have B= Nice to have C= Not really needed now
- Get the best project team you can. They may not be within 50 miles of your office. People who have done this before. Many times. You may have to pay more for an experienced team. Find a way to test team members before the live project starts.
- Get the right tools and use them. Examples: Fast computers, fast internet, BaseCamp for project management.
- How will you avoid reinventing the wheel? Leverage as much technology components as possible.
- Ready to focus your time/dollars on your real value add. (Unique technology, Market exclusive, essential service, IP, etc...)
- A plan to split up the project tasks. One developer may not be able to do everything. Parallel development.
- You have built a project contact list... Emails, phone numbers, addresses. Team members and bosses.



## Development Phase 1 – Startup

1. Precise documented deliverables and deadlines. (Every week or two)
2. Build a project contact list... Emails, phone numbers, addresses. Team members and bosses.
3. Clear understanding of what each person on the team “owns”.
4. Be ready to change your A, B, and C priorities based on estimates. Do not delay essential and/or difficult components until the end.
5. Be sure your A, B, and C priorities take into account potential revenues. User needs, revenues, and profitability need to drive service/product priorities.
6. When working with subcontractors – get a clear understanding of deliverables, dates and payments.
7. An understanding of how vacations, holidays, employee turnover and burnout will affect your schedule.
8. Avoid reinventing components that are already available somewhere. Focus on your companies real value ad.
9. An understanding of what tools and dependencies could affect the project deadlines.
10. Clear written agreements signed.
11. A Plan B. (Just in case...)

## Development Phase 2 – Live Project

1. Plan testing resources for deadline days. Use “fresh eyes” (new testers) where possible.
2. Provide quick accurate feedback on testing results and other comments. Use pictures. And provide exact steps to repeat bugs.
3. Test on several different systems and Internet speeds.
4. Plan for if/when deadlines are missed.
5. Be ready to change by your A, B, and C priorities. (Customer needs, Revenues & Profitability)
6. The Project Manager should be the primary interface with the development team. All non-developers should funnel comments through the project manager.
7. Keep resources like pictures and videos in organized files.
8. Sound the alarm if deadlines/deliverables are missed for any reason.

## Development Golden Rules

“Good design costs the same as bad design... And sometimes good design costs less.”

“If you don’t have a Plan B, you don’t have a Plan!”

“Every missed deadline is a significant data point. Take missed deadlines very seriously!”

“Empower any team member to stop a delivery anytime for any reason.”

## Development Phase 2 – Live Project – WARNING CHECKLIST!

- ❑ Missed deadlines and/or partial (incomplete) deliveries.
- ❑ Specification creep – Adding things to the design.
- ❑ Silence – Lack of communication.
- ❑ Project breakdowns (problems) not being addresses.
- ❑ Lack of task ownership.... Excuses instead of action and results.
- ❑ Testers who do not provide details for their findings.
- ❑ Key assumptions not working. Example: a tool or resource does not do what is expected or a key programmer goes to another project.
- ❑ Team members not using the project management/communication system. Must use something like BaseCamp. Email just does not work.

## Development Phase 3 – MVP Delivery

- ❑ Generally the last one to three weeks -> Rapid delivery, review, comments, revisions, priority changes, and approvals.
- ❑ Be ready with extensive testing resources. All hands on deck!
- ❑ Be ready to change your A, B, and C priorities. Maybe add a few of the “B” items back in if there is time and budget available.
- ❑ A Delivery Pit – Get the programmers, testers, designers tightly connected to minimize the feedback loop.
- ❑ Test everything two more times after you think it is done. Late modifications could have caused problems.
- ❑ Always backup at least three versions. Grandfather, father, son....

## Development Phase 4 – Post-Project Planning

- ❑ Plan for if/when deadlines are missed.
- ❑ Create a post delivery roadmap. Determine next delivery date, specification, team, and budget.
- ❑ Be ready for comments from users. Listen!
- ❑ Mend fences. Review budget/actual expenses.
- ❑ Delivery can be intense rebuild team.
- ❑ What were project surprises? Lessons learned? (unexpected events)



## HOW TO DEVELOP A PRODUCT ROADMAP

1. Keep it in 30 day chunks. (Maybe even two week chunks.)
2. Must be user needs/profitability driven...Not technology driven.
3. A collection of A, B, and C priority features/functions/benefits. Be sure at least one “exciting” feature is in each release.
4. Get tight firm time/cost estimates for each feature item.
5. Be sure to budget for maintenance and cleanup.
6. Layout each release on a timeline with features/functions for each delivery. Be sure everyone in the organization signs off on the roadmap.
7. Start development of the next release version as soon as each version is delivered.

## What is a specification, and how do you create one?

- Created from your prototypes. All details in one document.
- A good specification takes time and testing.
- Use pictures.
- Work out all the details in the specification to make things clear for your programmers. And Testers. And Users.
- Include all possible options. Be specific. Think it through.
- The more time you take on the specification the more time you will save during the expensive development phase.
- Can be as simple as an MS Word Document or created on something like MockPlus.

## More Project Management Tips and Tricks

- Set firm “red flag” dates – When to Panic?
- Delivery pit – Get everyone (programmers, testers, graphics and marketing people) together for the final delivery push.
- Watch for breakdowns – Assumptions not proving out.
- Watch for feature creep –Additional features...
- All testing reports should be step-by-step, repeatable. Pictures!