



Startup Boot Camp Roles / Team Formation

Engineer / Developer – Total tickets 32

- Visionary/Architect. Idea generation, shape features, repositioning, market fit, competitive landscape, research.
- Hardware designer – board level design (hardware, firmware, fpga, etc.)
- Mechanical designer – packaging, thermal design
- Lead SW Developer(s). Uses web framework, creates functionality; knows popular SW environments such as C++, Python/Ruby, Javascript, AJAX, Flash, HTML, databases, etc.
- Webmaster. SEO, analytics, domain registration, site hosting.

Biz Dev / Marketing / Sales – Total tickets 16

- Business Case, Strategy, Competition, Pricing model,
- Early adopter sales
- Graphic Artist. Color coordination, logos, icons, image libraries, etc.
- Content Creator. User-facing documentation, populate/organize wiki, design tutorial, usage studies.
- Marketer. Evangelism, blogging, advertising. Promotion Social media
- Manager. Coordinates all team member activities.

Finance / Accounting / Legal – Total tickets 8

- Lawyer. Business setup, guidance, law interpretation.
- Financing – pro forma income statement – business model

Operations / Manufacturing – Total tickets 8

- Production models, supply channels, cost modeling
- Project Manager

Teams

1. Most startups fail because of a failure of the team.
2. "Strong personalities" can derail the team
3. Specific roles should be decided upon, and then adhered to - Project Manager, Marketing, Strategy, Architect etc.
4. Give someone explicit authority over the "difficult decisions" - a "CEO" - don't run the team as a democracy
5. The person with the original idea does not necessarily need to be the CEO
6. Beware of things being too harmonious and congenial - it probably means you haven't asked enough questions. Early conflict is GOOD for the team.
7. Beware of leaders emerging without discussion from the type A personalities simply because they wanted to be leaders more than because they had the best experience or the most on the line.
8. Discover team-members motivation/expectations for participating in the weekend, as this will shape future interactions

9. The most important decision you'll make is which team to be in.
 1. **Don't choose an idea, choose a team:** Even the brightest idea will not get far if there isn't a good team behind it. Make a good compromise between an interesting idea and a good team. Make sure your team has enough back-end & front-end developers, as well as graphic designers if you're building a web application.
 2. **Think twice before joining a large team:** (More than 10-15 people) These teams are like large companies, they are not agile. It's harder to coordinate, reach consensus and make decisions. You'll also be working on a very small part of the whole, which might not be as exciting.
 3. **Make sure that others are committed:** All participants should share the same vision about the product. If you see people throwing in ideas that will take the product to a different direction, it's a bad sign. Try to anticipate if people will still want to work on this idea the next day.
 4. **Hold up to your ideals:** There will be boring ideas with huge business potential. Unless your only goal is to earn money, don't be fooled by them. Life is too short to work on boring ideas!

Strategy

1. Get an elevator pitch early in the process to help bind the team's vision
2. Get explicit consensus on the value proposition from the team
3. "Launch or Bust" by the end of the weekend.
4. Make something that people want and can grow organically - the ability to grow by itself without a sustained business development or marketing effort.

Scheduling and Scope Management

1. Set milestones. No really, set milestones (and keep them)

2. Control the scope aggressively - have someone act as Product Master (or else Business development, marketing, and user experience will all be telling dev what to build)
3. Development has to be in sync with compartmentalized tasks

Idea Pitchers

1. Follow the supplied pitch template and be ready to “sell” your idea to the attendees.
2. If you’ll build a web app, have a skeleton app that includes common functionality (authentication at least). Set up the infrastructure before the event (set up code repository and project management site etc.)

Equity and Legal Considerations

1. Projects have failed in the past when people have fallen out over ownership after the event
2. One solution is to agree that no ownership rights (IP or equity) accrue to any participant as a result of the Weekend