



Team Charter Template / Guide

Project charter for this team. It describes how we will work together, and how we will resolve conflicts.

Roles

Roles may represent individual team members or a team member may have multiple roles. This is a non-exhaustive list, but illustrates the main archetypes you would have in a team.

- | | | |
|---|-------------------|--|
| 1. Team Captain / Representative | | Reza Abasaltian |
| 2. Project Manager | | Sridhar Mothe |
| 3. Subject Matter Experts (SMEs) | Business: | Nidhi Kacchawa |
| | Machine Learning: | Gaurav Koradiya |
| | Lead Mentor: | Daniel Hooman |
| 4. Business Members | | Reza Abasaltian, Gaurang Vora |
| 5. Hardware Engineering Members | | Mohamad Hani Slim |
| 6. Software Engineering Members | | Gaurang Vora, Shilpi Mishra, Mohamad Hani Slim |
| 7. Design Members | | Rhitt Chakravarthy, Vaibhav Kulkarni |
| 8. Presenters/Media | | Vaibhav Kulkarni |

Principles

1. We work as a team. This means that sometimes a decision may not go your way.
2. When you commit, commit: If you say you are going to do something do it. Everyone has times where things come up etc. but if this happens raise with the team as early as possible.
3. Respect each other and the roles each team member is filling. Appreciate each team member's role and expertise. If you're a developer, recognise the expertise of the designer and visa versa.
4. Listen to consider, not respond. *[Some teams may want to institute a virtual "talking stick" if it gets too crazy]*
5. Team will have "stand ups" (Zooms/Meets/etc) at [set your times here] to check in.
6. If you are blocked on a problem for more than 30 minutes, you will let the team know.
7. We will check in with our mentor at [times here]
8. [Add more here]

Infrastructure:

1. Our communication channel will be [#job-creators](#)
2. Our documents will be stored here <https://github.com/abasaltr/Job-Creators>
3. Our technical assets will be located here <https://github.com/abasaltr/Job-Creators>
4. [Add more here as required]

Conflicts: (change to meet your needs)

Conflicts can and will happen. Our process to resolve them is as follows:

1. Seek expertise first. In technical matters, our technical team will provide recommendations. In design, our design capable members will provide recommendations. As yourself: Do other members of the team have more expertise in an area than yourself?
2. The project manager should have authority to set tasks. In some instances, where items become time-critical, the project manager may have to request a cut-down option that may be sub-optimal to meet deadlines.
3. The team captain has the final responsibility for a decision where a tie-break is needed.
4. [Add more here]

Basic Schedule / Template - **GUIDE ONLY**

Note that team members should be communicating - no silos. When it says "one group does this" and "another does that" this is not in isolation. Keep coming back and checking in.

| | |
|--|---|
| Day 1 - Hackathon Begins Form Teams & Share Ideas. Review the challenges, pitch your ideas and finalise your team. Navigate your way around and become familiar with the online workspace. | <ol style="list-style-type: none"> 1. Team Forming. Confirm you have the right mix of team 2. members. Set your team's SOP/expectations. 3. Start conceptualising the idea. 4. Review a basic schedule - try to set up meetings with mentors and SMEs early. Also check to make sure you get the right mentor (e.g. Tech, design or business) for the right domain. |
| Day 2 - All Day Design & Build | Day <ol style="list-style-type: none"> 1. Draft Business Model Canvas 2. Focus on the customer problem - validate |



| | |
|--|--|
| <p>Design, build and validate a proof of concept (model, MVP or prototype) using startup-inspired methods. Technical and Challenge Mentors can provide ad-hoc assistance on #ask_a_mentor channel.</p> | <ol style="list-style-type: none">3. Engineering / Design teams start to conceptualise and experiment. Go low-fi4. Validate low-fi (pictures/paper/diagrams/stories) with SMEs and stakeholders5. Refine BMC - Validate the concept with SMEs <p>Afternoon/Night</p> <ol style="list-style-type: none">6. Engineering / Design teams to build POCs/Prototypes etc.7. Presentation team should create a basic timeline / run sheet of the video8. Business members to work to describe / document Late afternoon / early evening is really your last chance to pivot if you really have to. Hopefully by this time, any adjustments will be minor. |
| <p>Day 3 - Final Day</p> <p>Submissions typically by 7pm Day 3.</p> | <p>Morning</p> <ol style="list-style-type: none">1. FINALISE Engineering / Design teams to build POCs/Prototypes etc. Do not try and pivot on Day 3. This group in the team might want to sleep :-)2. Work ramps up and you commence run throughs of the presentation and related deliverables3. Video presentation and post-production (if required) <p>Afternoon</p> <ol style="list-style-type: none">4. Final video presentation completed ready for upload Late Afternoon / Early evening.5. Leave yourself plenty of time to upload your final deliverables. <i>Ye hackathon gods are fickle.</i>6. Always give yourself a buffer of a couple of hours - aim for finalisation by 5pm. |

Best Practices / Common Pitfalls to avoid

- Get to an idea quickly and don't get lost debating whose "idea is the best" to pursue. Use objective measures to weight your decision including judging criteria, validation from target market, expertise in the team to execute quickly. The quicker you choose an idea to pursue, the more time you have to validate and build something awesome.
- Ask open questions to validate a user's needs rather than validating your idea or solution. You want to ask questions about their life, their challenges and what they've tried to solve this problem in the past.
- Done is better than perfect. Build something (anything) that you can demonstrate works,



before you drive for pixel perfection.

- When pitching, consider telling a story that inspires rather than simply walking through a solution's features. While it's useful to demonstrate how something works, using a narrative to communicate your why can be very compelling.
- [\[Add more here\]](#)

Best Practice Tips and Tricks for forming Teams in SLACK

- Start creating discussion threads within existing channels and forming your own locked or public channels with smaller groups
- Let people know what your passionate about working on and any ideas for a potential project in the relevant channels - you don't need to do a full reveal, but you should be sure to say enough about what you would like to do / what you need to get potential teammates excited
- Check out **#speed-teaming** for teams to join or create new ones. We've also created challenge specific channels that you can find potential team mates who are interested in solving problems you care about:
- Check out the **#resources-readings** and **#project-essential-tools** for project management and collaboration software tools for working with your teams

“Coming together is a beginning. Keeping together is progress. Working together is a success.” **Henry Ford**

This living document will serve as a North Star for your team or project. It will articulate your team's mission, the scope of operation, objectives, and commitment.

Roles

Roles may represent individual team members, or a team member may have multiple roles. It would be a non-exhaustive list but illustrates the main archetypes you would have in a team.

1. **Team Captain / Representative** - Represents the team, acts as a final decision point when there is a conflict.
2. **Project Manager** - Manage the team on the schedule
3. **Subject Matter Experts (SMEs)** - People who know the problem domain intimately from a customer (or stakeholder) perspective.
4. **Business Members** - Business modeling, financial analysis, and validation
5. **Hardware Engineering Members** - Building solution
6. **Software Engineering Members** - Building solution
7. **Design Members** - Responsible for user experience design
8. **Presenters/Media** - Responsible for the delivery of video and presentation elements

Principles

Teamwork does not happen on its own. It needs to be catalyzed, becoming part of hackathon culture, integral to people and processes. Once that happens, this competition will become more fun, productive, and creative.

- The Hackmakers(virtual hackathon specialists) will work as a portal for people from all of the different skill levels and approaches to meet and work mutually toward common goals. It is obvious that every other participant will not be having the same opinion. But working together respectfully is always appreciated.
- When participants present their projects at the end of a hackathon, we encourage the teams that didn't reach their goals to speak about what didn't work and why. Embracing failure as an opportunity to learn is part of what empowers our work culture.
- While this document endures primarily to prevent bad behavior, we also believe that our community members should work towards a higher standard.
- **Every contribution matters!** When different people share their knowledge and skills, they give a project a more elevated chance to succeed. When a participant shows up with other different skills, we look for ways to fit them into the team rather than reasons why it wouldn't work.
- After submitting an Entry, you should not submit copyrighted content protected by trade secrets or otherwise subject to third party intellectual property rights or other proprietary rights.
- And if you are the owner or have permission for your right content to post, it should be free from any harmful code or virus.
- Work to secure that the community is well-represented in all stages of development. Find those who are under-represented, and remove restrictions to access. Listen to



consider, not respond, and give full attention to all ideas, even if they seem unlikely at first.

Basic Schedule / Template - **GUIDE ONLY**

Note that team members should be communicating - no silos. When it says “one group does this” and “another does that” this is not in isolation. Keep coming back and checking in.

| | |
|--|--|
| Day 1 - Hackathon Begins Form Teams & Share Ideas. Review the challenges, pitch your ideas and finalise your team. Navigate your way around and become familiar with the online workspace. | <ol style="list-style-type: none">5. Team Forming. Confirm you have the right mix of team6. members. Set your team's SOP/expectations.7. Start conceptualising the idea.8. Review a basic schedule - try to set up meetings with mentors and SMEs early. Also check to make sure you get the right mentor (e.g. Tech, design or business) for the right domain. |
| Day 2 - All Day Design & Build Design, build and validate a proof of concept (model, MVP or prototype) using startup-inspired methods. Technical and Challenge Mentors can provide ad-hoc assistance on #ask_a_mentor channel. | <p>Day</p> <ol style="list-style-type: none">9. Draft Business Model Canvas10. Focus on the customer problem - validate11. Engineering / Design teams start to conceptualise and experiment. Go low-fi 4. Validate low-fi (pictures/paper/diagrams/stories) with12. SMEs and stakeholders13. Refine BMC - Validate the concept with SMEs <p>Afternoon/Night</p> <ol style="list-style-type: none">14. Engineering / Design teams to build POCs/Prototypes etc.15. Presentation team should create a basic timeline / run sheet of the video16. Business members to work to describe / document <p>Late afternoon / early evening is really your last chance to pivot if you really have to. Hopefully by this time, any adjustments will be minor.</p> |
| Day 3 - Final Day | Morning |

| | |
|--|---|
| <p>Submissions typically by 7pm Day 3.</p> | <ol style="list-style-type: none"> 7. FINALISE Engineering / Design teams to build POCs/Prototypes etc. Do not try and pivot on Day 3. This group in the team might want to sleep :-) 8. Work ramps up and you commence run throughs of the presentation and related deliverables 9. Video presentation and post-production (if required) <p>Afternoon</p> <ol style="list-style-type: none"> 10. Final video presentation completed ready for upload Late Afternoon / Early evening. 11. Leave yourself plenty of time to upload your final deliverables. <i>Ye hackathon gods are fickle.</i> 12. Always give yourself a buffer of a couple of hours - aim for finalisation by 5pm. |
|--|---|

Conflicts-

Participants and team members should prepare themselves for hackathon inventions before the event to avoid sticky legal and other technical issues. Here is what you can do to defeat disputes at hackathons.

- Create a comfortable, safe, and non-discriminatory event for a diverse set of participants. Having a well-defined set of rules helps in case of any issues cropping up during the event.
- To sustain a welcoming environment, formulate policies to deal with all kinds of harassment and lack of respect for others' opinions, and remind all attendees that it is up to them to make the experience incredible.
- As a participant, read the hackathon agreements and rules carefully before attending the event. Most of the time, the IP will belong to you. However, in case you work on a company's API, the idea will belong to it. Be sure to cross-check with the organizer and read the participation agreement to check for any indemnification clause.
- Communicate the code of conduct via the website, and slack and any other published material sent to sponsors, participants, speakers, hosting team members, and make people acknowledge their agreement to the terms and conditions.
- All attendees need to understand what behavior is to expect from them.
- Track your reporting policy or channel mentioned in the document to whom contact and how. Train your team before the event and select some people specifically to deal with any crisis and escalate.
- All outrages should be kept private, if possible, and handled impartially by following a clearly defined chain of custody.



- Decide what the outcome of the breach of conduct will entail for the person.
- We all are humans, and human makes mistake- try to not to judge each other

Getting ready for the event (overcoming the pitfalls)

- Try using organization tools to track all aspects- technical participation, etc.
- Choose the finalists from the idea submissions; get the best projects ready for the day of the hackathon.
- Finalize your speakers.
- Prepare your schedule.
- You need to get permission from the copyright holder in the reproduction of source code.
- Creative Commons licenses are a way for an author to allow the general public to “copy, distribute, and make certain uses of their work.
- Get to an idea quickly and don’t get lost debating whose “idea is the best” to pursue. Use objective measures to weigh your decision like judging criteria, validation from the target market, expertise in the team to execute quickly. The quicker you choose an idea to pursue, the more time you have to validate and build something awesome.
- Ask open questions to validate a user’s needs rather than validating your idea or solution. You want to ask questions about their life, challenges, and strategies to solve this problem in the past.
- Done is better than perfect. Build something (anything) that you can demonstrate works before you drive for pixel perfection.
- When pitching, consider telling a story that inspires rather than walking through a solution’s features. While it’s useful to demonstrate how something works,