Module 7

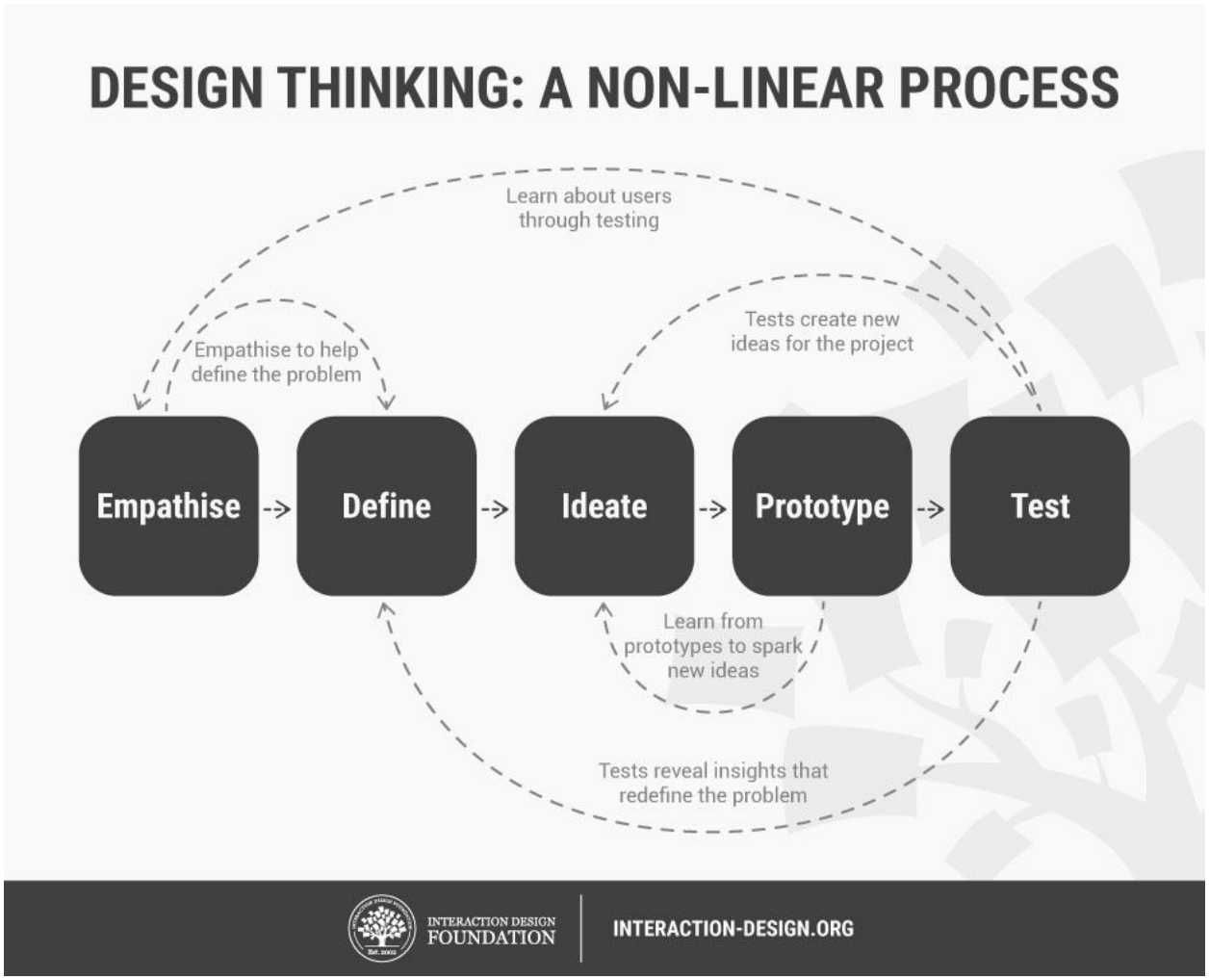
Congrats you now know the basics of programming and are ready to hit the ground running for the Innovation Fellowship.

However TKH is not just about showing you programming, our goal is to utilize technology to economically empower our community. This is done through helping you all secure successful employment. However, it is also important that you utilize your skills to solve problems for your community.

Gaining skills in this area is powerful and as the late great Stan Lee has told us “With great power comes great responsibility.” Because of this, it is important to understand that programming does not exist in a bubble, its power comes from allowing ordinary people to solve extraordinary problems. Those who have the skills to solve these problems impact what the solutions are and more than that what problems get solved in the first place. This is why our community's representation in the tech sector is important from coders working on new features for big companies all the way up to technical founders at companies who aim to solve problems in our community.

That being said, it is important to understand how to think about solving problems and how to involve the people you are looking to serve from the very beginning.

Design Thinking



As we think about solving problems, there are several key questions we must ask:

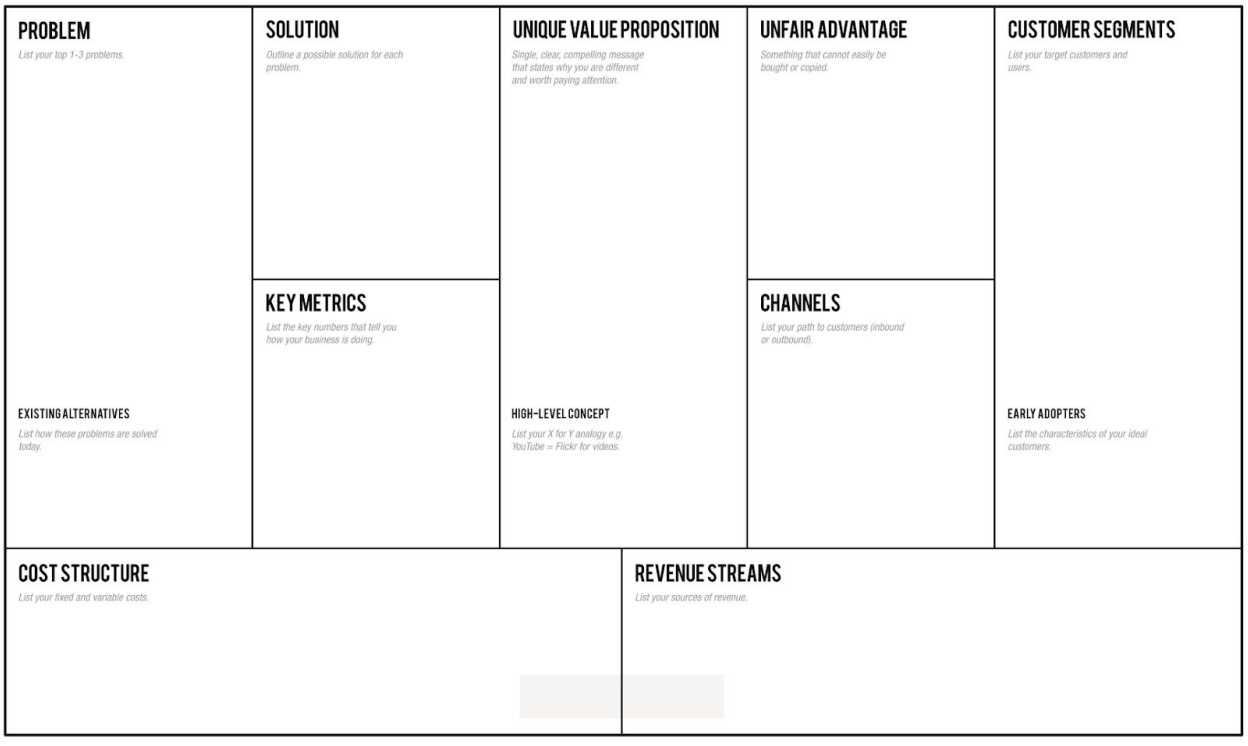
1) **Empathise:** What is the effect of the problem? What are the pain points? Are they time based financially or other? What about the current situation creates this negative effect? Who does this hurt the most? Are there multiple groups? If so, which group is affected the most?
2) **Define the Problem:** What is the underlying source? Once we understand the effect the current situation has on the lives of those we want to serve, we need to do the research, potentially interviewing people in the affected industry/group about the problem and their thoughts on where it is coming from. It may also be necessary to conduct a power mapping exercise. Once we

identify the people, situations, and/or factors that contribute to the root cause of our issues, we can start thinking about what our solutions and goals are.
    a) Are we looking to tackle the root cause?
    b) Are we looking to directly address a single or multiple pain point(s) of the problem?
    c) Are we looking to solve the problem directly or are we trying to provide information/aid in an action that our users can take to solve the problem themselves?
3) **Ideate:** Once you identify the problem you want to solve and the aspects of the problem you want to tackle, the next step is to brainstorm possible solutions. This is why no matter what aspect of a problem you are looking to attack, it is important to understand what aspects of the problem cause the actual pain points because at whatever level you are working, the solution must be able to address at least one of those pain points thoroughly. There are hundreds of Ideation techniques such as Brainstorm, Brainwrite, Worst Possible Idea, and SCAMPER. Note that a good solution addresses one thing well. It is better to have a single purpose app that does what it does in a well thought out way than have an application that has multiple purposes, as this can confuse users.
4) **Prototype:** Once you have your core idea, it is important to map out your MVP solution.
    a) What is an MVP? MVP (in this context) stands for Minimal Viable Product. Basically, it is a stripped down version of a solution that does only what is necessary to solve the problems without the bells and whistles.
    b) Why an MVP? When solving problems, time may be of the essence and keeping features to a minimum helps get solutions out faster. In addition, we often don't know what additional features users may want, that is why getting a product out fast and testing it with actual users is super important.
    c) MVP prototypes, whether they are functioning apps or interactive wireframes, should be tested internally first and then rolled out to potential users for testing.
5) **Testing:** Once a prototype is completed and testing begins, users will be tasked with using the app, raising issues about

usability, understanding how the solution solves the problem (if it misses a major pain point for example), and suggesting new feature sets for future releases.

NON-LINEAR PRODUCT DEVELOPMENT: Once testing is done, it is important to understand that the product development phase is not over, and as long as the product is being maintained and used, it will never end. New feature requests require new design sprints, covering all the stages above. Sometimes multiple teams will conduct the same processes for different sets of features at the same time.

Lean Business Model canvas: One of the best ways to visualize the process of thinking of an idea is to map it out using tools like a lean business model canvas.



The lean business model structure is usually for for-profit businesses but literally can be used to map out the solution for

anything you may be trying to create.

It is basically a way of organizing the process of design thinking:

Problem Definition: What is the problem you are looking to solve?
Customer Segments: Who experiences this problem and is likely to use your solution (Who is this for?)?
Solution: What is our MVP solution to this problem?
Key Metrics: How will we know if our solution is effective?
Unique Value Proposition: What makes our solution different from other solutions solving the same problem?
Channels: How will we get our solution into the hands of those that can use it? Will it be downloaded/is it free? Will you need to train people on how to use it? Etc.
Unfair Advantage: Is there a factor that makes your solution more effective than other solutions by the nature of a structural advantage either of the product or of your team (for example, do you have access to data that others don't?)?
Cost structure: What is the base cost it takes to create and maintain your solution? What is the cost of serving one person? The 100th person? The 1000th person?
Revenue Structure: How will your solution make revenue so it is able to fund itself and consistently provide the service?

Final Task:

Define a community problem you want to solve with technology and the people who it affects that you want to serve.

Write a full description of your application and bonus points for including drawings(digital or physical).
Here are some free wireframing tools.

Lastly
Fill out a lean business canvas with all the necessary factors.

This can be submitted in the module 7 folder through git.