Project Sunshine

Week 1 - team meeting: 8th February 2021

## Agenda

1. Housekeeping
2. Team members, processes & ways of working
3. Decide on tools & who to set up what
4. Creation of product design
5. Creation of Minimum Viable Product Documentation
6. Preparation for first sprint planning with company

## Notes to Agenda

*Tasks to be completed by Thursday 11th February 2021 in red*

### **1. Housekeeping**

* Team name
* Team members contacts - how to share with each other:
  + Github account
  + Email
  + Phone number
  + LinkedIn
  + Maybe twitter? For some team promotion work!
* Slack channel
* How to share Personal profiles (lumina)

### **2. Team members, processes & Ways of Working**

* Agree on the team processes and ways of working e.g following scrum or Kanban, arranging daily stand-ups, sharing people's availability

### Meetings

Availability:

<https://drive.google.com/file/d/1d-CEicpLT1xxD8qoUfyapB1Y919mZEr8/view>

* Fortnightly
  + 1 Sprint planning with companies: Thursday 11th February 5-6pm
  + 3 Sprint reviews with companies: Thursday 25th February, 11th March, 25th March 5-6pm
* Weekly
  + Weekly internal planning meeting - decide day & time - Mondays at 4pm?
  + Weekly internal review - decide day & time (e.g. Thursdays - same time as fortnightly meeting/& hour in between?
* Several times a week/daily
  + 10-15 minute standups - decide day & time
  + Some/all team members
* Specific this week:
  + Session to discuss & create design mockup - decide time & attendees
  + Session to create MVP
  + Run through preparation for meeting with company

### People

* Areas of experience
* Areas of interests/ improvements
* Roles you are looking to form after the programme
* How would you like us to support you
* How do you like to receive feedback?

### Roles

* Scrum Master
* Project manager

<https://medium.com/the-value-maximizers/scrum-master-vs-project-manager-an-overview-of-the-differences-73104d0264ab>

* note-taker/action recorder at planning meetings
* Code review - also see re. Github
* Backend
* Frontend
* System/ Business Analyst

### Ways of working

TR slides on agile:

<https://techreturners.github.io/agile_and_processes/001_introduction/lecture/#1>

* Scrum, Kanban, Agile
* Internal sprint lengths?

### **3. Tools**

* Slack
* Project management tools:
  + Trello
  + Jira
  + Github project board & GitHub issues
  + Github repository for documents, e.g. previous agendas, meeting notes (recommended by TR), designs, plans, useful resources
* Meeting tools
  + Zoom (40 minute limit)
  + Google Meet
  + Does Slack have the live call option on this version (paid version)
  + others?
* Co-working/sharing tools
  + Google Drive - current documents
  + Liveshare
* Mockup -> wireframe -> prototype
  + Options
    - Pen and paper
    - Balsamiq - <https://balsamiq.com/wireframes> (free trial for 30 days; looks good & relatively easy to use)
    - Figma <https://www.figma.com/> (good for collaborating; can add colours)
    - Adobe XD - <https://www.adobe.com/in/products/xd.html> (Popular in industry, free trial, can choose level of fidelity & presumably increasel)
    - Invision - <https://www.invisionapp.com/> (free for small teams)
* Github organisation
  + Tasks
    - Create Github organisation & invite other members
    - Create empty frontend Github repository
    - Create empty backend Github repository
    - Create frontend & push to main
    - Create backend & push to main
  + Agree on GitHub workflow
    - Chunking v. Trunk-based: <https://www.toptal.com/software/trunk-based-development-git-flow>
    - Pulling into master - minimum no. of team members to check?
    - Agree on commit message style

### **4. Creation of product design**

* produce designs for your application that demonstrate some of the user journeys through the application:
  + <https://classroom.google.com/c/MjY3ODI3Njk5NDM2/m/MjU0MzAzODI3Mzkw/details>
* Decide on tools - see above
* Arrange when to meet
* Parallel designs then joint or single joint design
* How high-fidelity do we want to go?
* UX - user involvement with a prototype (not wireframe stage)

### **5. Minimum Viable Product Documentation**

* Discuss and produce any Minimum Viable Product documentation (MVP) you have created around what will/will not be in your MVP:
  + <https://www.cayenneapps.com/blog/2014/11/25/5-steps-to-building-minimum-viable-product-with-story-mapping/>
  + <https://classroom.google.com/w/MjY3ODI3Njk5NDM2/tc/MjY3ODI3Njk5NDQ3>
* Need separate session - see above for arranging
* Starter document on Google docs based on TR recommended article: <https://docs.google.com/document/d/1lTJ-Kkqai5Je82WazAn7-llJN6pwdHdyDv7gEfqJgQI/edit#heading=h.tbl83kvydvkj>

### **6. Preparation for first sprint planning with company**

* Prepare for sprint planning session with sponsor: <https://classroom.google.com/w/MjY3ODI3Njk5NDM2/tc/MjY3ODI3Njk5NDQ3>
* Tasks to do - separate session - see above for preparing
* Create agenda in time to send to company before the meeting
  + <https://docs.google.com/document/d/1K2TOcZNpzX2vdgVCg3bdomcqLJUMgWu8IZEXOQTiS7w/edit> (first draft - TR notes)
* Collate visuals
* Practice/Agree who is going to say what
* Think of questions for the company
* Decide who will take which questions

## Future tasks to add to project management tools

* Backend Structure
* Division of labour, taking into account what we each want to learn & moving project forward
* Languages, technologies

## Techreturners Project Management Tips

* It's a good idea to decide on your workflow with your team so everyone knows the process
* Split work into discrete chunks ahead of time - make sure you all know what's coming up next and what the right order of work should be
* Maybe designate someone as project manager to ensure the process is adhered to, and take on the lion's share of organisation?
* You may want to have regular stand up meetings (remote or in person) where each team member spends a couple of minutes updating the rest of the team on their progress and explaining what they plan to do next, and whether they have experienced any problems.
* Break all work into the smallest chunks possible so you are always merging really small changes into the main branch. It helps with the merge conflicts, trust us
* It's worth creating a GitHub repository for your project documentation.

**Aluns Notes**

**Project Checklist**

1. Decide on project name

2. Consolidation of project points generated from each person

3. Quick discussion on people’s strength areas

a. programming languages

b. project management

c. general project experience

d. Anything else?

4. Decide on project tools

a. JIRA

b. Trello

c. Anything else?

5. Decide on roles

a. project manager

b. person leading stand-ups

c. discuss the options to swap roles throughout the project

d. Anything else?

6. Identify focus areas for each person

a. Backend

b. Frontend

c. Anything else?

7. Identify for each person new learning requirements

a. E.g., to learn some frontend in the project in addition to their focus areas

b. Anything else?

8. Identify any learning material we could use

a. google classroom areas with links to YouTube video and other documentation

i. e.g., introduction to React page in google classrooms

9. Plan structure for the week

a. daily stand-ups

b. availability

c. Frequency of meetings

d. Anything else?

10. Decide project methodology

a. Scrum

b. Kanban

c. Others?

11. Sprint review structure discussion

a. https://classroom.google.com/c/MjY3ODI3Njk5NDM2/m/MjY3ODI3Njk5NTE3/detail

b. Plan and structure sprint planning and sprint review session

i. 1 sprint planning session every 2 weeks

ii. 3 sprint reviews every 2 weeks

iii. \*\*Sponsor attends every sprint review session\*\*

iv. Discuss the brief

v. Discuss 1 or 2 ideas about an application we could build

1. Include how the application might be used

2. Include what the application will be used for

vi. Discuss roles and responsibilities

vii. Think of questions to ask sponsors (shared amongst the team)

viii. Each person should introduce themselves and their background for every sprint review session

ix. Explain brief/goals to the sponsor

x. Identify order of who is talking and when and what about

xi. Visuals – pen and paper drawings and demos of the application and photos of the task lists

xii. Q and A at the end of each session

1. Identify who will answer what type of questions

xiii. Prepare questions to ask the sponsor about the brief and who will ask what

xiv. Duration of sprints

xv. Date and times for sprint review and planning sessions

xvi. Minimum for first session we need ideas about what your thinking to create (e.g. pen and paper ideas, and\or document of items we are considering)

c. Plan and structure sprint review sessions

i. Demo work

ii. Gather feedback on what team has completed

12. Discussion on if we should introduce paired programming

a. If so, identify which pieces of work to do this for.

b. Others?

13. Setup GitHub repository

a. <https://github.com/techreturners/ProjectTipsAndWorkflow>

b. Do some practice commits, merges etc on a zoom call

14. Identify which technologies and languages to use in the project

a. Frontend

i. HTML, CSS, REACT, JAVASCRIPT, NODE.JS

b. Bootstrap

i. open-source CSS framework directed at responsive, mobile-first front-end web development. It contains CSS- and JavaScript-based design templates for typography, forms, buttons, navigation, and other interface components

c. Backend

i. Java, C#, JSON, AWS

d. Database

i. SQL, mySQL

e. API

f. Supporting services

i. Amazon RDS

ii. Serverless framework

iii. Terraform

iv. Circle CI

g. Responsive web development

i.

h. Scripting languages

i. Python

i. Any cloud technologies?

j. Anything else?

15. Discuss technical delivery approaches

a. TDD

b. Systems testing

c. Integration testing

d. Peer code reviews

e. Anything else?

16. Decide how and where we will host our application

17. Decide on GIT version control approach

a. Git Flow

b. Trunk Based Development

c. Decide on naming convention for GIT commits

18. Decide on development tools to use

a. VS Code

b. IntelliJ

c. Postman

d. Git

e. Circle CI

f. Anything else?

20. Discuss and produce Minimum Viable Product

a. https://www.cayenneapps.com/blog/2014/11/25/5-steps-to-building-minimum-viable-product-with-story-mapping/

b. We need to deliver the smallest set of features that delivers customer value

i. Story mapping

ii. Step 1 - Capture the primary goal of your product

1. Define primary goal

iii. Step 2- Define the main process in the product

1. Think less about features and more about the tasks

2. Add process stages to map

iv. Step 3 – create a list of features for each stage

1. Add to map

v. Step 4 – prioritize the features and rearrange on the map (high value priority at top)

1. Question 1: *How important is this feature for finishing the process?*

2. Question 2: *How often will the feature be used?*

3. Question 3: *How many users will use this feature?*

4. Question 4: *How much value will the feature bring to the customer?*

5. Question 5: *How risky is this feature?*

vi. Step 5 – define the MVP

1. Draw horizontal line (essential features above the line and other features below the line). Above the line = MVP

21. Wireframing and Prototyping

a. <https://classroom.google.com/c/MjY3ODI3Njk5NDM2/m/MjU0MzAzODI3Mzkw/details>

b. <https://www.justinmind.com/blog/low-fidelity-vs-high-fidelity-wireframing-is-paper-dead/>

c. https://www.justinmind.com/blog/20-inspiring-web-and-mobile-wireframe-and-prototype-examples/

d. Produce designs for the application early on in the process

i. Produce prototype designs

1. Basic sketches

2. Low-fidelity wireframes

a. paper based

b. shows basic content and does not show user interactions.

c. One piece of paper per web page

d. Quickest way to get feedback and should be used very early on

e. Tools: MS Office suite, Justinmind

3. High-fidelity wireframes

a. computer based

b. show user interactions

c. Tools: Justinmind

4. Other tools

a. Balsamiq - <https://balsamiq.com/wireframes> (Free trial offered)

Figma <https://www.figma.com/> (Great for collaborating on a design)

Adobe XD - <https://www.adobe.com/in/products/xd.html> (Popular in industry, free trial)

Invision - <https://www.invisionapp.com/> (free for small teams)

ii. Make a plan of when to show prototypes to sponsors in order to gather quick feedback. Do this before coding starts.

iii. pen and paper

22. Learn about responsive design and grid layout

a. Build in project plan time for this as there is at least one lengthy video on this.

b. Resources:

i. <https://techreturners.github.io/yrtt_intro_to_frontend/lecture/#1>

ii. <https://techreturners.github.io/yrtt_intro_to_frontend/>

iii. <https://github.com/techreturners/yrtt_responsive_design_playground>

iv. <https://getbootstrap.com/>

v. https://storage.googleapis.com/your-return-to-tech/curriculum/session\_recordings/week\_002\_SessionA\_ResponsiveDesign\_FINAL\_PROCESSED.mp4

23. Architecture overview

a. <https://storage.googleapis.com/your-return-to-tech/curriculum/architecture_overview.mp4>

b. Create -

c. Read -

d. Update -

e. Delete -

24. Build project plan

a. Work package creation with dates and times for delivery

b. Include when to hand in assignment 1

**Assignment 1**

1. Create a GitHub organisation

2. Agree on the project management tools and systems that your group will use (such as JIRA, Trello, GitHub issues)

3. Produce some initial sketches of the application you might create (low tech, pen and paper is fine!)

4. Discuss and produce any Minimum Viable Product documentation (MVP) you have created around what will/will not be in your MVP

5. Agree on the team processes and ways of working e.g following scrum or Kanban, arranging daily stand-ups, sharing people's availability

=======

Submission Process

=======

1. Please submit a link to your GitHub organisation

2. Screenshots of any project management tools you've used

3. Any support documentation and links such. It's worth creating a GitHub repository for your project documentation.

**Key Google classroom areas to visit**

1. Introduction to React

a. #### Intro (video):

<https://www.youtube.com/watch?v=dGcsHMXbSOA>

#### Brilliant Intro to React Heathers just found! ✨

<https://www.learnwithjason.dev/let-s-learn-react>

#### React documentation:

<https://reactjs.org/docs/getting-started.html>

1. Architecture overview (Access Passcode for Recording: %a\*9#RXr)

a. <https://storage.googleapis.com/your-return-to-tech/curriculum/architecture_overview.mp4>

b. https://techreturners.github.io/yrtt\_introduction/002\_overall\_architecture/lecture/index.html#1

2. Agile processes & tools

a. <https://techreturners.github.io/agile_and_processes/001_introduction/lecture/#1>

b. https://us02web.zoom.us/rec/share/2whxtsul84Q3hOn7flr9kndiu2oNyG\_n\_PnJaJFNlPeMF\_88EejbFuIdA2R6M3Hg.rHtz1kPp\_bc53\_qI

3. Git Flow vs Trunk Based Development

a. https://www.toptal.com/software/trunk-based-development-git-flow

4. Good Commit Messages

a. <https://chris.beams.io/posts/git-commit/>

b. https://www.freecodecamp.org/news/writing-good-commit-messages-a-practical-guide/

5. Introduction to your team projects

a. <https://github.com/techreturners/ProjectTipsAndWorkflow>

b. <https://drive.google.com/file/d/115U7MQ-k0rTCBGQaBautQm3XRv6zv4U9/view>

6. Tech for Good Brief – Zuhlke

a. <https://drive.google.com/file/d/1WQOPtu8Spv0kIBDtOXigjgCYDBxV57Mn/view>

7. Minimum Viable Product

a. <https://www.cayenneapps.com/blog/2014/11/25/5-steps-to-building-minimum-viable-product-with-story-mapping/>

8. Wireframing and Prototyping

a. =====

Blogs to support you:

=====

<https://www.usability.gov/how-to-and-tools/methods/prototyping.html>

<https://www.justinmind.com/blog/low-fidelity-vs-high-fidelity-wireframing-is-paper-dead/>

====

Tools you might want to explore:

=====

Pen and paper (well a sharpie, designs always look better with a sharpie ha)

Balsamiq - <https://balsamiq.com/wireframes> (Free trial offered)

Figma <https://www.figma.com/> (Great for collaborating on a design)

Adobe XD - <https://www.adobe.com/in/products/xd.html> (Popular in industry, free trial)

Invision - <https://www.invisionapp.com/> (free for small teams)

9. Responsive Design and Grid Layout

a. <https://techreturners.github.io/yrtt_intro_to_frontend/lecture/#1>

b. <https://techreturners.github.io/yrtt_intro_to_frontend/>

c. <https://github.com/techreturners/yrtt_responsive_design_playground>

d. <https://getbootstrap.com/>

e. https://storage.googleapis.com/your-return-to-tech/curriculum/session\_recordings/week\_002\_SessionA\_ResponsiveDesign\_FINAL\_PROCESSED.mp4