



Managing Weekend Projects





/TABLE OF CONTENTS



/01

**Why do we need a
managing system?**



/02

First steps



Design doc, time
allocation

/03

Creating tasks



Main tasks, breaking
down tasks,
connecting tasks,
prioritizing

/04

**Project
management**



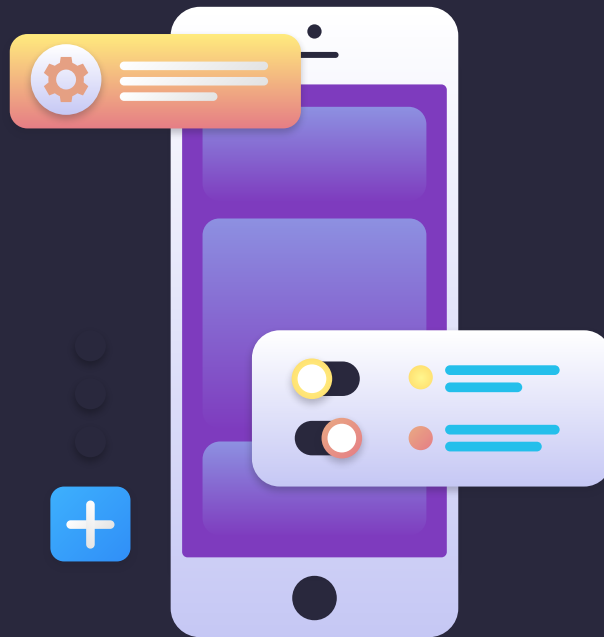
Tools: git, github
issues, time
tracking, trello





/01

Why manage projects?



Why manage projects?



- Working on a software project is a big task.
- Whether you are alone or a part of a team, have a simple project or a complicated one – there are a lot of moving parts.
- So what can we do in order to create and manage our projects more easily?



Why manage projects?



- In this lecture we will talk about things that can help you manage your weekend project better.
- In real life you might do things a bit differently, because you will have more team members, different goals, etc.
- The methods I will talk about are a part of the AGILE family.



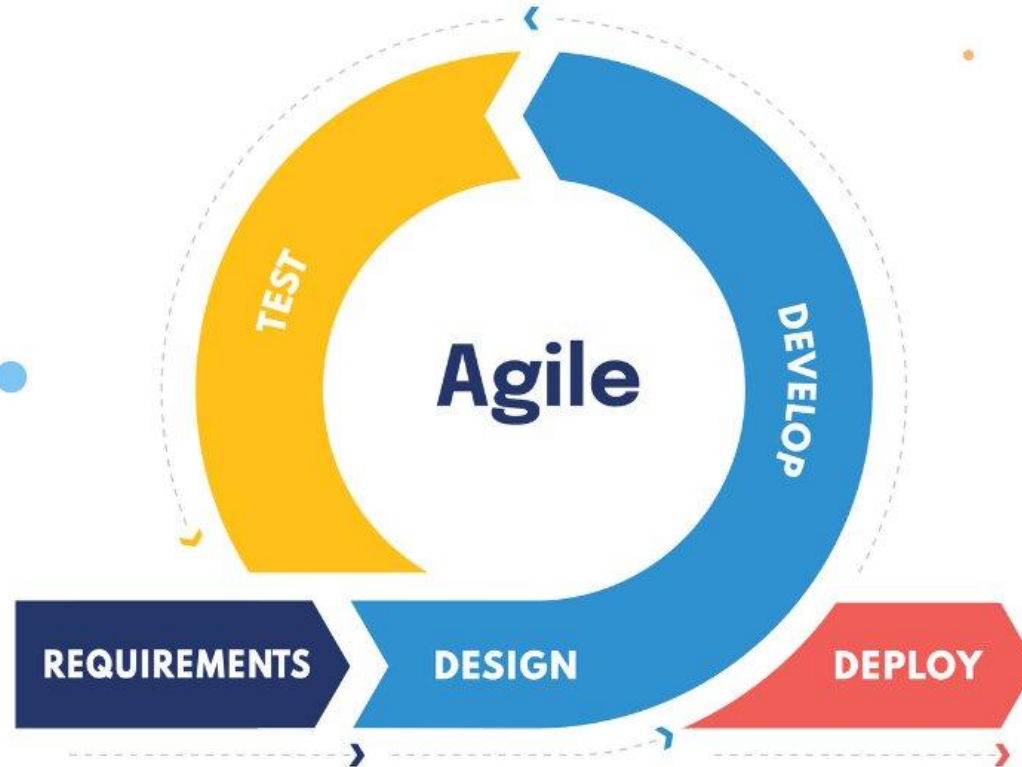
Why manage projects?



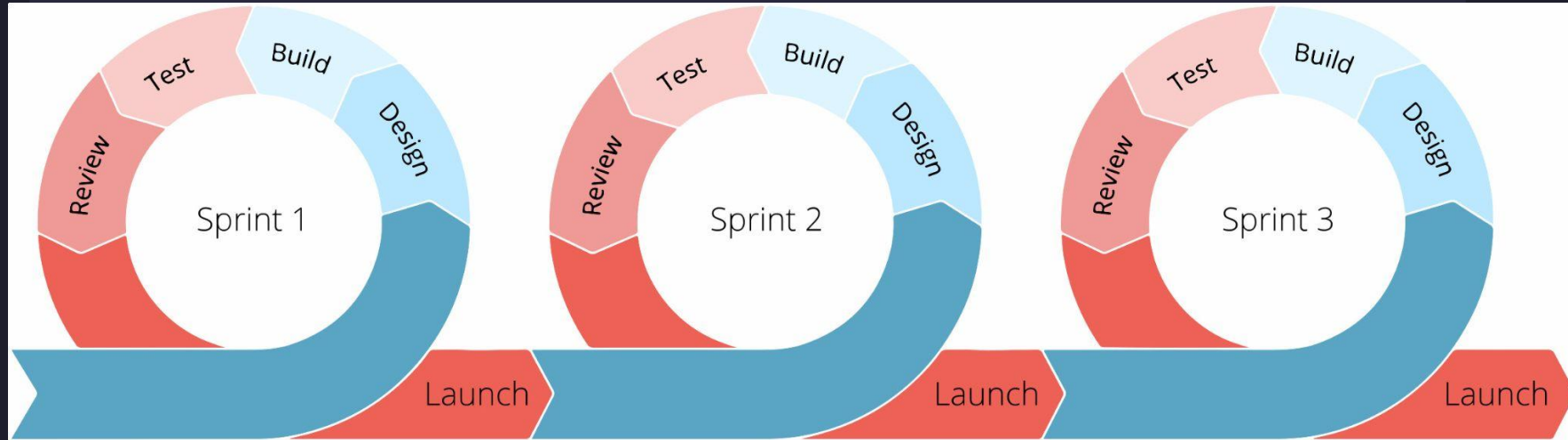
- The idea of the agile methodology is that software development is not a linear project.
- It's not start → finish after X hours, it's more like iterations:



Why manage projects?



Why manage projects?



Why manage projects?

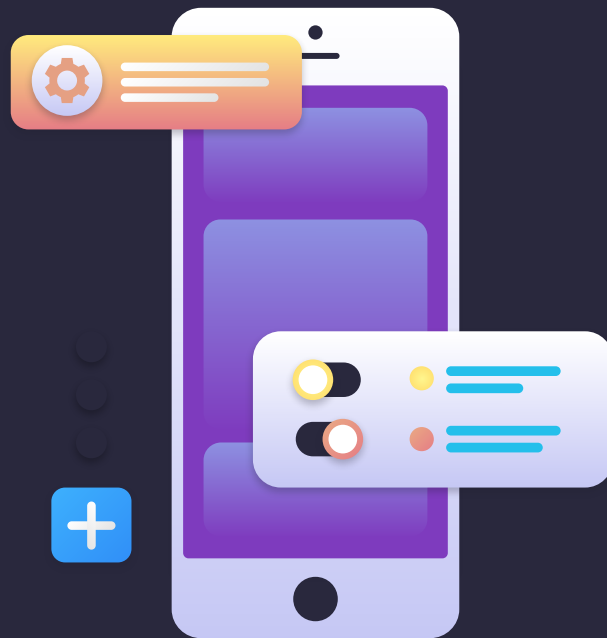


- In this lecture we will talk about things that can help you manage your weekend project better.
- In real life you might do things a bit differently, because you will have more team members, different goals, etc.
- So the way I talk about agile in the weekend project sense is not exactly how it works in real life.
- I will describe points from SCRUM.



/02

First steps



First steps



- First of all, there are no rules!
- Everyone works differently, and it's ok. **The only important thing is that the job gets done, in a practical and clean way.**
- If some or all of what I say doesn't help you achieve this goal - then don't use it.



First steps



- Managing a project is a big field of study, and one in which i'm not an expert.
- I only give you a few tips and methodologies that I found helpful.
- I will give more resources that can give you a more professional advice on the subject, at the end of the slideshow.

First steps



- For me, the first step is the **design document**.
- This is the document that describes the project in the general sense.
- If you are a part of a business or a part of a big team, then your design doc will be different than what I am going to recommend.





First steps



- For a weekend project, I recommend that the design document will include:
 - **Project goal**
 - **Project design**
 - **Project total available time**
 - **Project milestones**





First steps



- **Project goal:** what are we trying to build?
- Give details, but not specifics.
 - Example: a landing page for tesla, that shows the latest modules.
 - Not: a page for tesla that shows model 3 and the max speed.



First steps



- **Project design:** how we are going to build it. What are the main components.
- Describe each component general purpose, but without too specific details.
- For example:
 - First page: general tesla module details
 - Second page: testa technical specs
 - Third page: contact page

First steps



- **Project time:** how much total time this project has.
- In real life, there not be a specific number here, but a general timeline.
- For the weekend projects, give the actual number of hours you have for the project.
- **Be exact!** Don't write general numbers. If you have 20.3 hours, this is what you write.



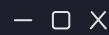


First steps



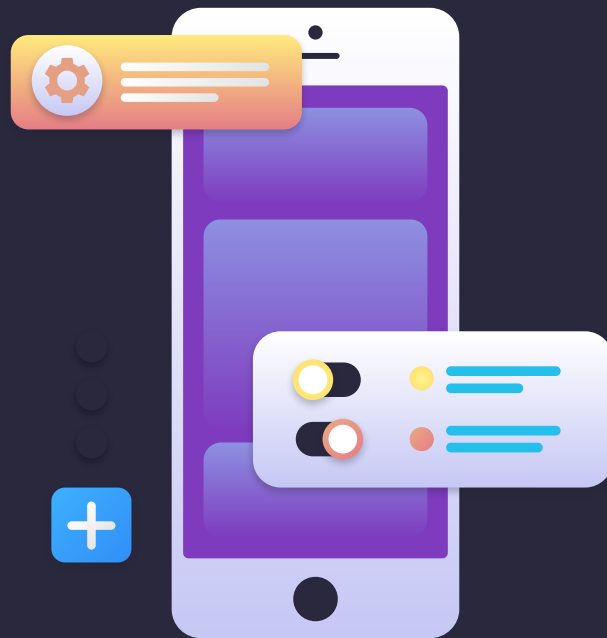
- **Project milestones:** we will talk about it more on the next section.





/03

Creating tasks



Creating tasks



- This is one of the most important parts of planning a software project.
- The first missions you need to create are the big one – the milestones.
- You can think on a mile stones as significant steps in the development process, such that there is a huge difference before and after the milestone.





Creating tasks



- For example, for tesla:
 - Milestone 1 - entire site grid structure.
 - Milestone 2 - add images and specific style
 - Milestone 3 - make app responsive





Creating tasks



- Now that we have a milestone, we need to break them down to specific tasks.
- Here is the place to be as specific as you can.
- I recommend that each task inside a milestone would have the following attributes:
 - Name of task
 - Description of task - definition of done (DOD)
 - Assignee (how is responsible = you)
 - difficulty/Estimated time
 - priority





Creating tasks



- For example:
 - Milestone 1 - tesla grid
 - i. Task A:
 1. Name: header
 2. Description: create header.
DOD: have a nav header with 4 elements inside
Make sure that the alignment is correct - for
outer elements and inner elements.
 3. Priority: high
 4. Difficulty : 4 / estimated time: 3 hours



Creating tasks



- What is difficulty?
- We can use a scale of 1-4 for difficulty.
- More difficult tasks take more time.
- Why not just estimate the time each task is going to take?
- Because it's hard to know beforehand.





Creating tasks



- Important:
 - Be as precise as you can with your tasks.
 - Make sure that not all tasks have the same priority.
 - Give detailed DOD. more details = easier to track your progress.



Creating tasks



- After we have tasks, it's time to assign time for each task.
- If we used the difficulty scale (1-4) for tasks, we need to estimate the ration points/hours.
- For example, I can estimate that 1 difficulty point = 1 hour.



Creating tasks



- Then, as the project continues, I can measure the time a task took and compare it to the difficulty point, and change the ratio accordingly.
- Important: you have to make sure that the **total amount of hours for each task is equal or smaller than the total amount of hours in the design document!**



Creating tasks

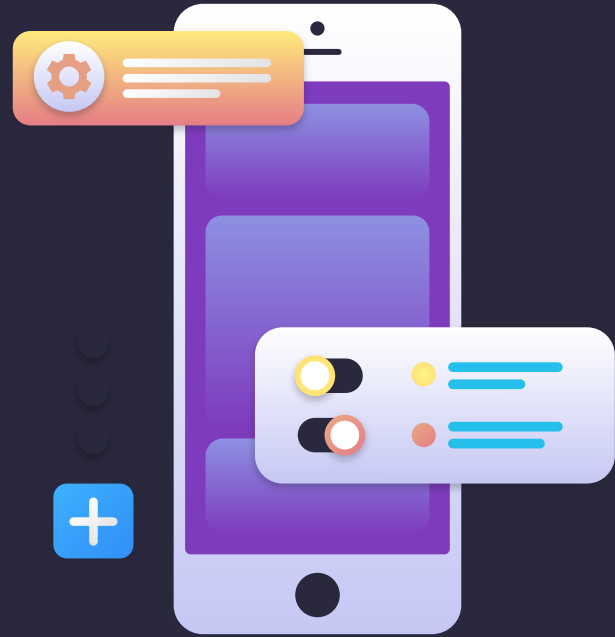


- So now we should have a good foundation for our project.
- We have clear goals for the project, with general time
- We have big tasks (milestones) that are broken down to smaller, well defined tasks.
- Each task has estimated time, and the total time is equal or smaller than the total time



/04 Tools for project management

→





Tools for project management



- Know our task is pretty simple: work on tasks based on priority, and track the time.
- There are millions of tools to help do both.
- I will go over basic ones.





Tools for project management



- Github projects:
 - Allows you to create tasks (issues), attach them to a milestone, and place them according to their status.
 - tutorial:
https://www.youtube.com/watch?v=YVFfa5VljCDY&ab_channel=edutechional





Tools for project management



- everhour:
 - Track time based on task. Can connect to github issues.
 - <https://everhour.com/>
 - Tutorial:
https://www.youtube.com/watch?v=kcrv_d-F0-c&ab_channel=KnowledgeByMarcus





Tools for project management



- Git and github:
 - Git can be used to break you big tasks into smaller branches.
 - Github issues and milestones can also be used





Tools for project management



- General project management software:
 - Jira
 - Monday
 - asana



Tools for project management



- Important: the tool is not important! It's just here to help.
- You can use google docs for everything if you want, the important thing is to complete the tasks.
- Tip: put tasks with time on google calendar. See visually how your days is going to look like. Track your progress! Make sure you are not overflowing!



Tools for project management



- Read more about agile:
<https://www.atlassian.com/agile>
- Agile methods: SCRUM, kanban, etc
- Scrum: <https://www.wrike.com/scrum-guide/>
- Kanban: <https://www.atlassian.com/agile/kanban>



Thanks, and good luck!

Contact us:

Yishai

yishain@appleseeds.org.il

0524605642

Shadi

shhadys@appleseeds.org.il

0543113297

