

T/R

Best Practices on Task Tracking

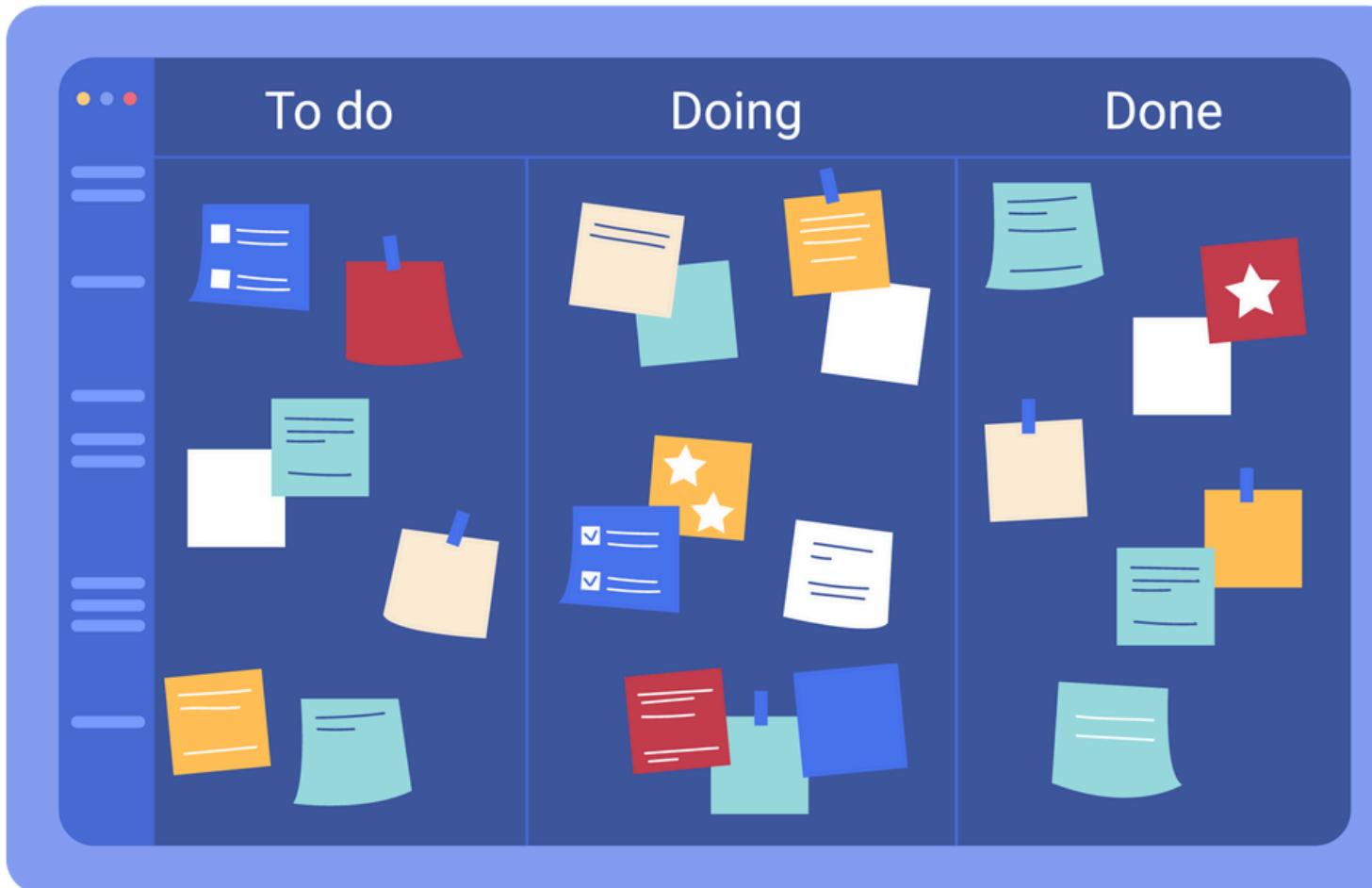
Africa To Silicon Valley



Can you explain what task tracking is?



Task Tracking



Task tracking is the process of monitoring and managing tasks throughout a project's lifecycle.

Task tracking involves assigning tasks, setting deadlines, and tracking progress for timely project completion.

Task tracking ensures organization, prioritization, and alignment with project goals and timelines.



What are the consequences
of **not** using a task tracking
mechanism?

Problems we aim to address

Here are some problems we are facing and how we are planning to over come them.

- 01 Missed deadlines, incomplete work, and lack of coordination
- 02 Lack of visibility
- 03 Difficulty in prioritization
- 04 Lack of accountability

How we are solving it

Here are some problems we are facing and how we are planning to over come them.

01

By providing a standard and consistent approach for task tracking across projects

02

The documentation includes guidelines on best practices on how to define tasks, assign responsibilities, set deadlines, and monitor progress.



Contents

01

TERMINOLOGIES YOU
NEED TO KNOW

02

THE BEST PRACTICES TO
FOLLOW

03

EXAMPLES



01

METHODOLOGY WE FOLLOW

Agile Methodology

What is agile?

Agile methodology is an iterative and incremental approach to software development that values **flexibility, collaboration, and rapid delivery** of working software.



01

METHODOLOGY WE FOLLOW

Agile Methodology

How does it Work?

Agile methodology works by breaking a project down into smaller, more manageable iterations or **sprints**, typically lasting 2-4 weeks. During each sprint, the development team works on a small set of **prioritized tasks** and aims to deliver a working software feature by the end of the sprint.

The cycle

Functionality - Planning - Implementation - Review - Delivery



Agile Methodology

01

METHODOLOGY WE FOLLOW

Here's a step-by-step overview

Product Backlog: The product owner creates a list of features and requirements, known as the product backlog.

Sprint Planning: The development team meets with the product owner to select a set of items from the product backlog to work on during the upcoming sprint.

Sprint: The development team works on the tasks in the sprint backlog, with the goal of delivering a working software feature by the end of the sprint.

Sprint Review: At the end of the sprint, the development team conducts a sprint review with the product owner and other stakeholders

Sprint Retrospective: After the sprint review, the development team conducts a sprint retrospective to reflect on the sprint and identify areas for improvement in the next sprint.



Then repeat

02

TERMINOLOGIES YOU NEED TO KNOW

Scrum In Agile Methodology

What is scrum?

Scrum is a framework for implementing agile methodology in software development. It is one of the most popular agile frameworks. Scrum consists of a set of **roles**, **events**, **artifacts**, and **rules**, which help teams to work together more effectively and efficiently.



02

TERMINOLOGIES YOU NEED TO KNOW

Scrum In Agile Methodology

Scrum roles

Product Owner: responsible for defining and prioritizing the product backlog, and making decisions about what to build.

Development Team: responsible for designing, building, and testing the product increment in each sprint.

Scrum Master: responsible for facilitating the Scrum process and ensuring that the team follows the Scrum framework.

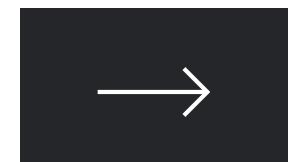


01

TERMINOLOGIES YOU NEED TO KNOW

Scrum In Agile Methodology

The cycle
Functionality - Planning - Implementation - Review - Delivery



Scrum Events

Sprint: a timeboxed iteration, usually lasting 2-4 weeks, during which the DT works to deliver a potentially shippable product increment.

Sprint Planning: a meeting held at the beginning of each sprint, where the product owner and development team collaborate to select and plan the sprint backlog items.

Sprint Review: a meeting held at the end of each sprint, where the development team presents the completed work.

Sprint Retrospective: a meeting held at the end of each sprint, where the team reflects on the previous sprint and identifies areas for improvement.

01

TERMINOLOGIES YOU NEED TO KNOW

Issue In Agile Methodology

What is Issue?

Issue refers to any problem, task, or activity that needs to be completed in order to deliver the product increment. It can be anything from a one on one to ACL and A2SV generative AI hackathon preparation.

The types include **sub-task**.



01

TERMINOLOGIES YOU NEED TO KNOW

Best Tool For Agile Development

- 01 JIRA
- 02 Github Projects
- 03 Asana
- 04 Clickup



02

THE BEST PRACTICES IN A NUTSHELL

Always plan your sprints in advance

Prioritize your issues

Break your tasks into subtasks

Use a definition of “DONE”



while creating issues, be as descriptive as possible

Allocate time daily to mark your progress on your Github project board

Estimate story points

Conduct retrospective at the end of the sprints

02

THE BEST PRACTICES IN A NUTSHELL

DON'T

Underestimate the power of having a consistent system/methodology to rely on

Miss sprint planning session and don't let it be a culture for your team members

Delay sprint start date or end date, and never miss sprint retrospectives

Assume people will understand what is expected out of a task

DO

Read the documentation and learn what we have to follow as a team

Be there for sprint planning and have all members join the session

Have your sprint planning, start, and end date on time and always have sprint retrospectives.

Always be as descriptive as possible to avoid confusions / miss-understandings



02

THE BEST PRACTICES IN A NUTSHELL

Things that should always be populated when creating a issue(task)

Summary (title)

Description

Acceptance criteria

Priority

Steps to do the task in a checklist

Assignee

Component or Label



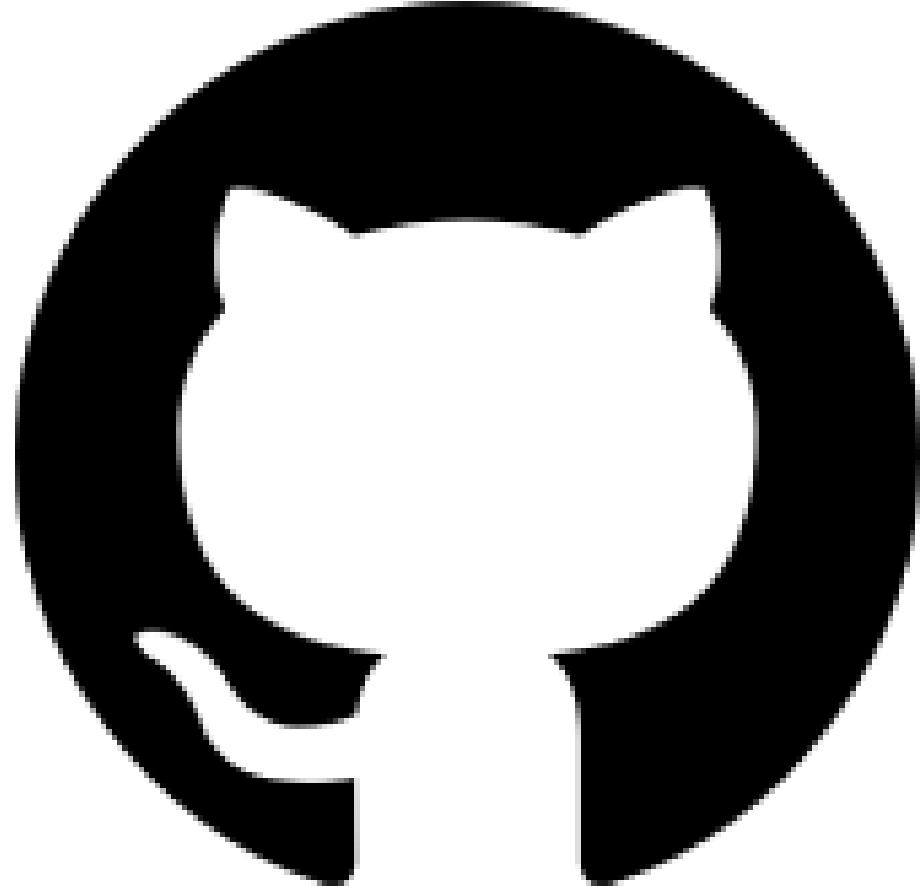
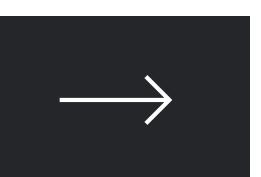
03

GITHUB ISSUES AND PROJECT PRACTICE TIME

Github Projects Practice Time

Let's get to know github project and how to implement some of the best practices.

[Open open the guide](#)



03

GITHUB PROJECTS PRACTICE TIME

GitHub Projects Practice Time

Let's get to know jira and how to implement some of the best practices.

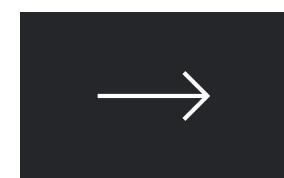
Example Project

Issue: Implement User Authentication System

Priority: High

Acceptance Criteria: Can anyone tell me what this is?

Planned steps: checkpoints in the form of subtasks



"Perfection is not attainable, but if we chase perfection, we can catch excellence."

Vince Lombardi



Africa To Silicon Valley

Q/A

Thank you!
