COMP 3550

10.2 — PROJECT MANAGEMENT TOOLS

Week 10: Measuring Team and Project Successes

GITLAB BOARDS (RECAP)

Check out that link ->

for lots of neat ways teams might use gitlab boards

Columns — represent workflow stages (e.g., To Do, In Progress, Review, Done)
Issues — track individual tasks, bugs, or features
Tags / Labels — categorize issues (priority, type, team)
Assignees — clear ownership for each task

How to use GitLab for Agile software development

How Agile artifacts map to GitLab features and how an Agile iteration looks in GitLab.

gitlab.com

JIRA (INDUSTRY STANDARD)

Issue Types

- Story — new feature or user need
- Task — general work item
- Bug — defect to fix
- Epic — large initiative broken into smaller stories/tasks

JIRA (INDUSTRY STANDARD)

Issue Types

- Story — new feature or user need
- Task — general work item
- Bug — defect to fix
- Epic — large initiative broken into smaller stories/tasks

Story Points

- Effort estimates (relative, not hours)
- Used for sprint planning and velocity tracking

Sprint Planning

- Select backlog items for the sprint
- Ensure estimates and acceptance criteria are clear
- Balance capacity with priority

JIRA (INDUSTRY STANDARD)

Issue Types

- Story — new feature or user need
- Task — general work item
- Bug — defect to fix
- Epic — large initiative broken into smaller stories/tasks

Story Points

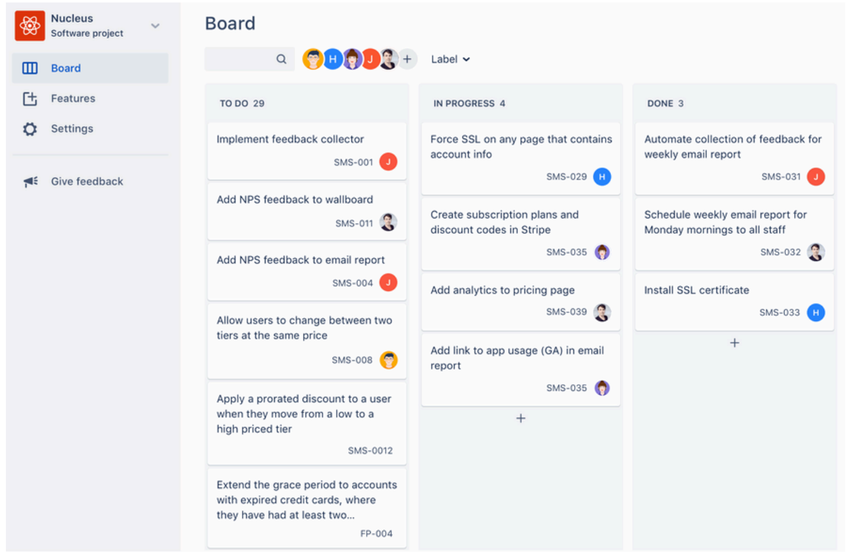
- Effort estimates (relative, not hours)
- Used for sprint planning and velocity tracking

Sprint Planning

- Select backlog items for the sprint
- Ensure estimates and acceptance criteria are clear
- Balance capacity with priority

Sprint Tracking

- Move issues across the board (To Do → In Progress → Done)
- Use burndown charts to monitor progress
- Daily stand-ups help surface blockers early



https://www.atlassian.com/blog/jira/preview-new-agility-boards-jira-software

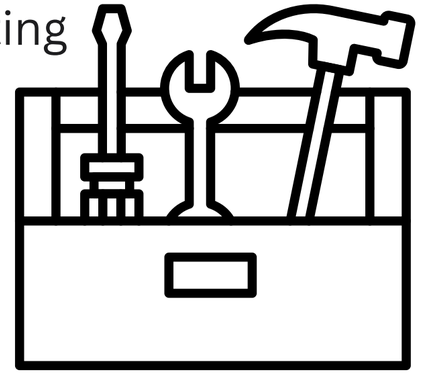
AN IMPORTANT REMINDER:

A board is only as useful as the updates you make — stale boards create false confidence.

CHOOSING THE RIGHT TOOL

Key Factors to Consider

1. **Team Size & Structure**
   - Small, tight-knit team → lighter tools (e.g., GitLab Boards, Trello)
   - Large, multi-team org → structured tools (e.g., Jira) with reporting & permissions
2. **Technical Needs**
   - Integration with version control, CI/CD, and deployments
   - Support for custom workflows, automation, and advanced tracking
3. **Project Complexity**
   - Simple project → minimal overhead
   - Complex product → features like epics, dependencies, advanced reporting
4. **Collaboration Style**
   - Async, distributed → tools with rich commenting & notifications
   - Co-located → simpler visual boards may be enough



AGILE COORDINATION BEYOND TASKS

More Than Just a Task List

- Project tools aren’t only for tracking work, they’re a shared brain for the team.

Communication

- Document decisions in comments (not just in meetings)
- Tag teammates to pull them into discussions
- Use issue history to understand why a change happened

Shared Memory

- Store key links, diagrams, and acceptance criteria inside issues
- Use labels/tags to make past work easy to find
- Keep sprint retrospectives and action items in the tool for reference

Benefits

- Reduces issues with knowledge that lives only in people’s heads
- Onboards new team members faster
- Creates transparency for stakeholders

AGILE COORDINATION BEYOND TASKS

More Than Just a Task List

- Project tools aren’t only for tracking work, they’re a shared brain for the team.

Communication

- Document decisions in comments (not just in meetings)
- Tag teammates to pull them into discussions
- Use issue history to understand why a change happened

Shared Memory

- Store key links, diagrams, and acceptance criteria inside issues
- Use labels/tags to make past work easy to find
- Keep sprint retrospectives and action items in the tool for reference

Benefits

- Reduces issues with knowledge that lives only in people’s heads
- Onboards new team members faster
- Creates transparency for stakeholders

AGILE COORDINATION BEYOND TASKS

More Than Just a Task List

- Project tools aren’t only for tracking work, they’re a shared brain for the team.

Communication

- Document decisions in comments (not just in meetings)
- Tag teammates to pull them into discussions
- Use issue history to understand why a change happened

Shared Memory

- Store key links, diagrams, and acceptance criteria inside issues
- Use labels/tags to make past work easy to find
- Keep sprint retrospectives and action items in the tool for reference

Benefits

- Reduces issues with knowledge that lives only in people’s heads
- Onboards new team members faster
- Creates transparency for stakeholders

AGILE COORDINATION BEYOND TASKS

More Than Just a Task List

- Project tools aren’t only for tracking work, they’re a shared brain for the team.

Communication

- Document decisions in comments (not just in meetings)
- Tag teammates to pull them into discussions
- Use issue history to understand why a change happened

Shared Memory

- Store key links, diagrams, and acceptance criteria inside issues
- Use labels/tags to make past work easy to find
- Keep sprint retrospectives and action items in the tool for reference

Benefits

- Reduces issues with knowledge that lives only in people’s heads
- Onboards new team members faster
- Creates transparency for stakeholders

PROJECT PAUSE & REFLECT

Think about your current project and your day-to-day life:

- What tools are you already using to stay organized?
- Do they actually help you — or are they just adding noise?
- How do they compare to what we've seen in class (GitLab, Jira)?

Explore Beyond the Course

Consider trying one new tool for personal or team productivity:

- **Todoist** — lightweight, recurring tasks, natural language input
- **Microsoft To Do** — integrates with Outlook, cross-device sync
- **ClickUp** — flexible project & personal task management
- **Notion** — documents + tasks + databases in one workspace

Reflection Questions

1. If you could only keep one tool, which would it be and why?
2. What's one pain point in your current workflow that a new tool might solve?
3. Could you apply an Agile principle (visibility, iteration, feedback) to your own personal organization?