

# COMP 3550

## 4.4 — AGILE IN PRACTICE: TEAM ROLES, STAKEHOLDERS & VELOCITY

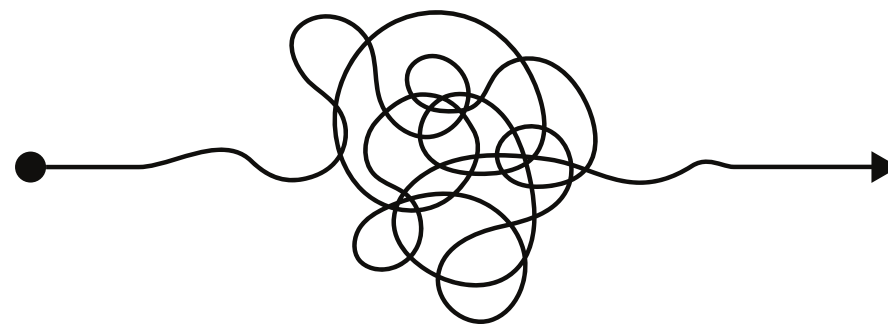
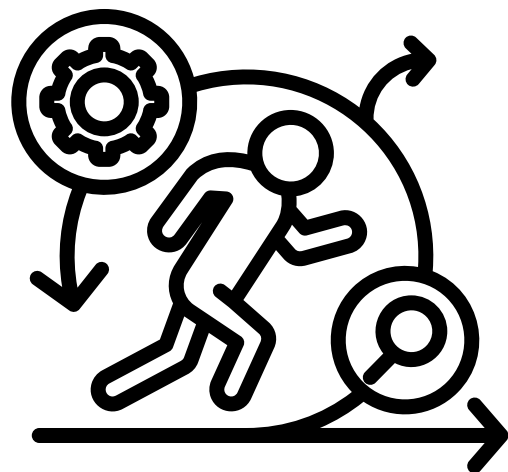
Week 4: Exceptional Testing &  
Technical Debt

# AGILE IN THE REAL WORLD

- Agile does not mean do whatever you want

# AGILE IN THE REAL WORLD





- Agile does not mean do whatever you want
- It's about fast feedback, not chaos



- **Not** a free-for-all
- **Not** “no planning”
- **Not** “no deadlines”
- **is** Small, testable chunks
- **is** Continuous communication
- **is** Frequent delivery, fast correction

# KEY ROLES (EVEN IN STUDENT TEAMS)

## Roles That Keep Things Moving:

-  Team Lead / Scrum Master
  - Keeps the group organized, tracks tasks, helps unblock people
-  Developer(s)
  - Write and test the actual code — core builders of the product
-  QA / Tester
  - Catches bugs, tests edge cases, gives fast feedback
-  Documentation / Git Upkeep
  - Tracks issues, writes READMEs, manages version control clarity

# STAKEHOLDERS & FEEDBACK

## Who's the Product Owner Here?

Your instructor

- Sets goals
- Reviews progress
- Gives feedback

## Weekly Check-ins = Sprint Reviews

- Show what you've built
- Explain what's not done yet (and why)
- Get quick, useful feedback to guide your next sprint

"Demo what you did, own what you didn't."  
Be transparent. Be proud of progress — even partial. That's Agile.

# IN YOUR PROJECT

## 2-3 Asynchronous checkins a week

- Daily standup is unrealistic and unnecessary for this project
- 2-3 asynchronous checkins help keep you accountable in between iterations
- What you did/what you will do

## Weekly Check-ins = Sprint Reviews

- At the start of each iteration, set a planning meeting

# TRACKING VELOCITY (LIGHTWEIGHT)

- A rough measure of how much your team accomplished this sprint (week)

How Can You Track It?

- Number of completed GitLab issues
- Merged pull requests (PRs)
- Commits that tie to meaningful progress
- Finished checklist items in your planning doc

*Keep It Simple, Just: “What did we finish this week?”*

**Tracking velocity helps your team plan better and spot blockers early — without adding overhead.**

# WORKING IN ITERATIONS

## Set Clear Sprint Goals

- Each week = a mini project
- Prioritize small, finishable tasks
- Aim to have something demo-able by check-in

## Always Leave Time to:

- Test your code
- Review PRs (code + comments)
- Refactor or clean up — not just build



# WORKING IN ITERATIONS

## Set Clear Sprint Goals

- Each week = a mini project
- Prioritize small, finishable tasks
- Aim to have something demo-able by check-in

## Always Leave Time to:

- Test your code
- Review PRs (code + comments)
- Refactor or clean up — not just build

## Avoid the “Merge Everything at Midnight” Problem

Last-minute chaos = bugs, burnout, and busted builds

Instead: merge early, merge often, and build in review time

# HEALTHY AGILE HABITS

- Communicate blockers early
- Don't take silent shortcuts
- Use MRs as communication, not just code dumps



## Day 1 – Kickoff

- Sprint Planning (in person or online)
- Define sprint goals + tasks
- Assign roles as required

# HEALTHY AGILE HABITS

- Communicate blockers early
- Don't take silent shortcuts
- Use MRs as communication, not just code dumps



## Async Check-In #1 (Early Week)

What you did / What you plan to do / Any blockers?

- Short post or message per person (GitLab, Discord, Slack, etc.)
- Keeps momentum going

# HEALTHY AGILE HABITS

- Communicate blockers early
- Don't take silent shortcuts
- Use MRs as communication, not just code dumps



## Mid-Sprint Work Period

- Start building core features
- Write tests early
- Pair up or review early PRs

# HEALTHY AGILE HABITS

- Communicate blockers early
- Don't take silent shortcuts
- Use MRs as communication, not just code dumps



## Async Check-In #2 (Midweek)

- Update on progress
- Reassign or adjust if someone's stuck

# HEALTHY AGILE HABITS

- Communicate blockers early
- Don't take silent shortcuts
- Use MRs as communication, not just code dumps



## Final Week – Test & Polish

- Refactor, test, and review
- Prepare documentation for check-in

# HEALTHY AGILE HABITS

- Communicate blockers early
- Don't take silent shortcuts
- Use MRs as communication, not just code dumps



## Async Check-In #3

- Quick “state of the team” wrap-up
- Flag anything that’s unfinished or rolled forward

# HEALTHY AGILE HABITS

- Communicate blockers early
- Don't take silent shortcuts
- Use MRs as communication, not just code dumps



## End of Sprint – Review

- Demo individual progress to each other
- Reflect as a team
- Plan next iteration



# PAUSE & REFLECT

**Before you move on, take a moment to think:**

- Which role on your team are you most comfortable with?
- Which role would challenge you to grow?
- How does your team handle feedback — are you getting enough of it, early enough?
- Are you planning your work with testing and review in mind?