ITI8740/ITX8522: Software Development Team Project. 2026

02. Team Formation. Teamwork

Martin Verrev
martin.verrev@taltech.ee


CHANEL
logitech

Today's Agenda

1. Presenting the ideas
2. Team formation
3. Teamwork Basics
4. Next steps

When speaking in public:

- Do not worry and be yourself
- Use simple words (no jargon)
- Look people in the eye, smile, and stay positive
- Use your body, tone of voice, and pauses for emphasis
- Speak loudly enough, and remember to breathe
- If something goes wrong, stay calm and keep going

Just practice a bit - easy!

You have the structure

1. Say **hello** and introduce yourself.
2. What's the **problem**?
3. What's the **solution**?
4. Explain the **special sauce**.
5. Is there **money** to be made?
6. Say **thank you**, smile, and listen for applause :)

Let's listen to the ideas

Think and discuss.

Team Formation.

For Each Team

- Minimum: 4 people
- Maximum: 6 people
- Decide what you want to do

Teamwork:

1. Understand yourself
2. Understand your teammates
3. Understand the task

Self-Awareness Is the Starting Point of Effective Teamwork

Self-Awareness means understanding oneself—our thoughts, feelings, and behaviors. For example:

- What working style suits us best
- What our strengths are
- What we need to improve

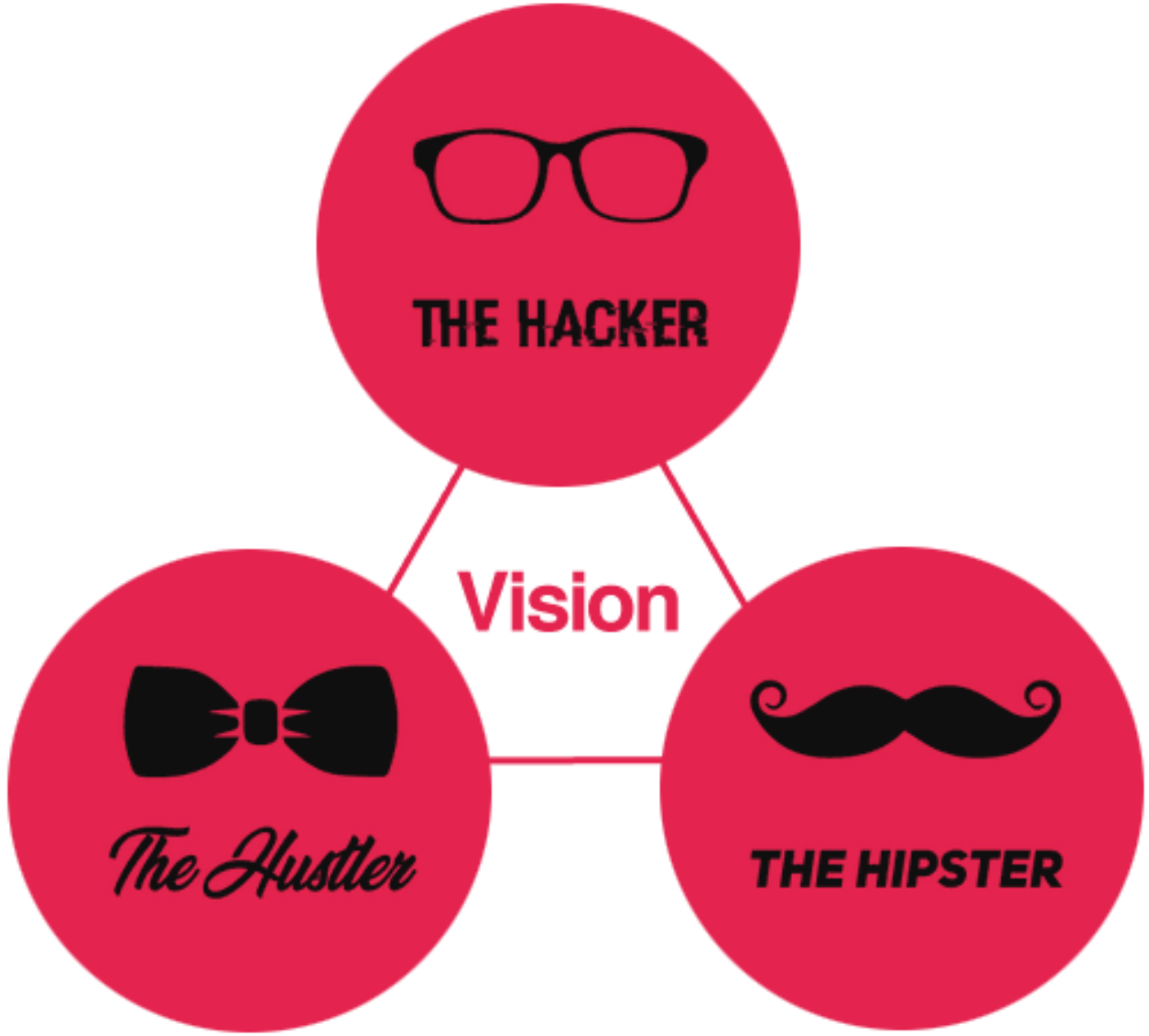
When we understand ourselves, we begin to see why we think or behave in certain ways. This makes working with others much easier.

Commitment

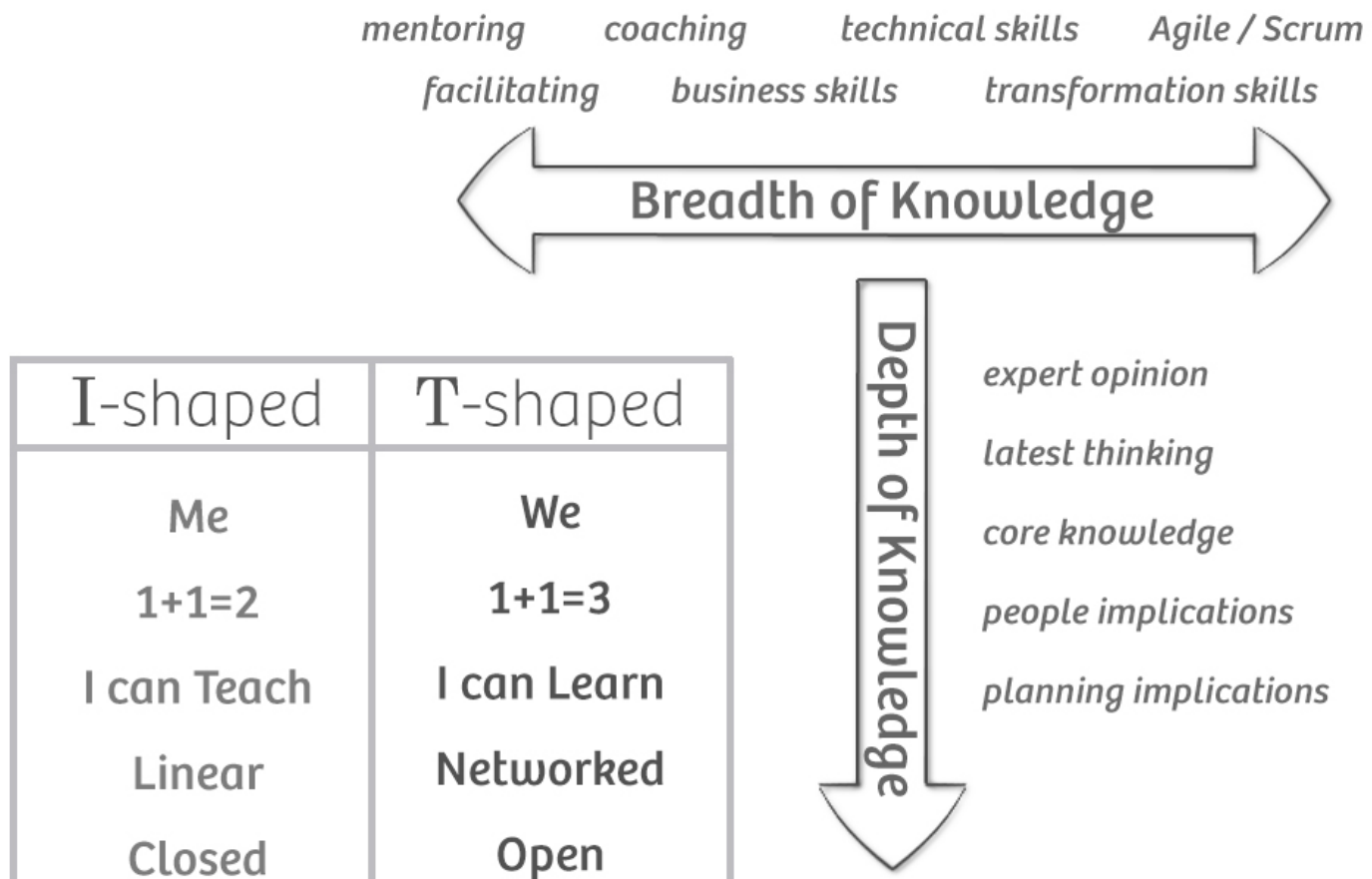
- Don’t let your teammates down
- Execution is the only thing that matters
- Keep your word

Divide Roles and Responsibilities

- Developers
  - Backend
  - Frontend
- Team Lead
- (Analyst?)
- (Public face / client relations / sales representative)


THE HACKER
Vision
The Hustler
THE HIPSTER

We want T-shaped People/Skills



I-shaped	T-shaped
Me	We
1+1=2	1+1=3
I can Teach	I can Learn
Linear	Networked
Closed	Open

T-shaped People/Skills

The concept is attributed to Tim Brown, the CEO of IDEO, a design and innovation company. IDEO is known for its emphasis on interdisciplinary collaboration, and the T-shaped skills concept emerged as a way to describe the kind of skills and mindset valued in such an environment

See also: https://agilecoffee.com/toolkit/t-shaped/

Commitment

Commitment Is Very Important

It should not be a chore—your passion, or lack of it, will shine through.

The less of a chore your app is to build, the better it will be. Keep it small and manageable so you can actually enjoy the process. If your app doesn't excite you, something's wrong. If you're only working on it in order to cash out, it will show. Likewise, if you feel passionately about your app, it will come through in the final product. People can read between the lines.

From: Running Lean, *Chapter 9_*
https://basecamp.com/gettingreal

Learning

- First, just try
- 15 min: Google
- 20 min: Ask
- Google some more
- Stack Overflow
- Also: GitHub, Reddit, and Hacker News
- LLMs

Teaching and Team Play

- Help others out
- Pair programming
- Be a team player; don’t be a jerk
- Spend informal time together

Communication

- Be proactive
- Be on the same page
- Try to help, think along, be constructive
- Manage expectations
- If something goes wrong, communicate

What Is the End Result?

- Analysis — an understanding with the customer
- Light vs. heavy analysis and documentation
- Just text vs. tools and methodologies
- Mockups vs. a prototype
- Paper prototyping (there’s an app for that)
- Fluid UI, etc.
- Output: task list

Plan

- Always prioritize
- Divide the project into smaller pieces
- Try to estimate the pieces
- Estimate work effort based on tasks
- Estimate calendar time
- Create a plan (the initial plan always fails)

What Does the Client Want?

Research and communication is very important.

Your understanding may change over time.

Feedback from Customers and the Supervisor

- The supervisor and customers must be involved
- Proactive communication
- Make sure the client reviews your work
- Stay on the same page
- Course-correct

Iterations

- A repeating cycle
- A work-and-feedback routine
- Focus
- Detect and learn from your mistakes

Choose Your Process

- Process frameworks: RUP, XP, and Scrum
- Following a framework by the book seldom works
- Weekly cycle – plan, build, review
- Daily cycle – stand-up / PPP

Personal Process and Routine

Everyone

Track your time and tasks

Developers

- Test your code
- Don’t commit broken code
- Review the diffs of others’ commits

Personal Process—Project Manager

- Project manager – feel the pulse
- Meetings – end with TODOs and commitments
- Follow up on commitments
- ALWAYS make notes
- Communicate with the client
- Adjust plans

Set Up Tools

Communication Channel

Create an instant-messaging chat in Teams

Code Repository

- Teamwork requires a code repository
- Set up GitLab

Task/Issue Tracker

- Use the GitLab issue tracker
- Set up a Kanban board: a simple `TODO/Doing/Done` board works.
- Track progress
- Make it visible to the client and the whole team

The Process Is Not God

- Modify the process during the project
- Discard practices that do not work
- React to problems and find better practices

Kick-off

- Start with a kick-off meeting
- Make sure everyone understands the process
- Divide tasks on a volunteer basis
- Start the development cycle

Project Agreements: Checklist

- Internal Communications Channel: Teams? Slack?
- Method: Kanban or something else
- Working days: when, where and how - both individual and with team. The total contribution for every team member should be 19hrs per week and tracebale via issue tracker
- Team meetings: when, where and how?
- Stand-ups: when, where and how?
- Retrospectives: When and how?
- Sprints: when and how?

Assignment for Week 3 (1/2)

- Hold a kick-off meeting with your team and decide which project you want to build.
- Create a project on GitLab: https://gitlab.cs.ttu.ee/. Name the project `ITI8740_26_<team_name>`.
- Join the course team channel if you have not already done so. See: https://courses.cs.ttu.ee/pages/ITI8740
- Create an initial project wiki containing at least:
  - A list of team members and roles; decide on a team lead
  - An initial description of the idea (maximum 0.5 A4 page)
  - A page with general project agreements (see the previous slide)

Assignment for Week 3 (2/2)

- Set up an issue tracker on GitLab.
  - Create a Kanban board with at least TODO / IN PROGRESS / DONE.
- Read Chapter 1, "Running Lean Roadmap," from the book "Running Lean." A free version can be found at https://danielpandza.wordpress.com/wp-content/uploads/2013/01/running-lean.pdf

HOW TO: Kick-off

Start with a kick-off meeting.

- Share team contacts
- Agree on workflow (see next slide)
- Decide on the project
- Make sure everyone understands the outcomes
- Agree on a realistic scope

  ...
- Divide tasks on a volunteer basis
- Start the development cycle

HOW TO: Project Agreements Checklist

- Internal communication channel: Teams or Slack?
- Method: Kanban or something else?
- Working days: when, where, and how—both individually and as a team. Each team member should contribute 19 hours per week, traceable through the issue tracker.
- Team meetings: when, where, and how?
- Stand-ups: when, where, and how?
- Retrospectives: when and how?
- Sprints: when and how?

Reading Materials

- Lean Validation Playbook: https://guides.co/g/the-lean-validation-playbook/60783
- Startup Playbook https://playbook.samaltman.com/
- Thoughtbot Playbook https://thoughtbot.com/playbook
- Meta Principles I Learned from Running Lean. https://medium.com/lean-stack/meta-principles-i-learned-from-running-lean-28df02489265

Thank you!

If No Ideas are found

- Choose a group leader
- Map out the ideas in the group (very quickly)
- Spend with each idea 15-20 min answering mentioned 5 questions from week 1 slides
- Somebody has to take notes on the answers.
- Choose 3-5 ideas to present.