TEAMWORK - A PRIMER

Christian Kaestner

GROUP PROJECT

- 4 Milestones and a Final Presentation
- Milestone 1: Build movie recommendation service (2 weeks)
- Milestone 2: Model and infrastructure quality (2 weeks)
- Milestone 3: Monitoring and continuous deployment (2 weeks)
- Milestone 4: Drift and feedback loops (1 week)
- Groups of 4--6 students

INFRASTRUCTURE

- We simulate 1M users on a platform with 160k movies
- We provide an underpowered virtual machine
- You can observe the system through logs streamed in Apache Kafka
- We send recommendation requests to your VM
- Technology choices entirely up to you
- You may use additional resources (e.g. AWS)

- Focus on reliable deployment; we do not care about accuracy

STEP 1: ESTABLISH COMMUNICATION AND MEETING PATTERNS

- Agree on how to communicate in the team: Email? Slack? Whatsapp?
- Agree on communication expectation. Different people have different habits and expectations. Be explicit!
    - Read emails daily? On weekends?
    - Respond to urgent chat messages within 3h?
    - Be available for chat during certain hours?
- Find meeting times: one early, one or two in the middle, one late? Extra meetings as needed?
- Write down expectations!
- Set realistic expectations: All have other classes and distractions; communicate availability openly

TEAM CITIZENSHIP

- Not everybody will contribute equally to every assignment -- that's okay
- But be good team citizen!

- Be responsive and responsible
- Stick to commitments, work on assigned tasks
- When problems, reach out, replan, communicate early, be proactive
- Come to meetings on time

TEAM MEMBER EVALUATION FORM

- Has the student attended team meetings?
- Has the student made a serious effort at assigned work before the team meetings?
- Has the student notified a teammate if they would not be able to attend a meeting or fulfill a responsibility?
- Does the student attempt to make contributions in group meetings?
- Does the student listen to their teammates' ideas and opinions respectfully and give them careful consideration?
- Does the student cooperate with the group effort?

(not asking for amount or quality of work completed)

TEAM COMPOSITION AND ROLES

- Team members have different strength and weaknesses -- that's good
- Make use of individual strength of team members
    - Split responsibilities and work, ...
    - Onboard, pair up, ...
- Consider assigning and rotating roles and responsibilities
    - E.g., coordinator (host), moderator, and scribe for team meetings, submitter responsible for final checks and submission of milestone
- See GitHub for example team policy

STEP 2: DIVIDING THE WORK

- Coordinate at meetings
- Read assignment before meeting
- Discuss big picture and how to divide work (inner teams?)
- Consider task dependencies

- **Write down explicit deliverables**
  - *Who* does *what* by *when*
  - Be explicit about expected results, should be verifiable
  - Track completion, check off when done
  - GitHub issues, Trello board, Google docs, ... -- **single source of truth, with history tracking**
- Complete deliverable list **during meeting**: everybody writes their own deliverables, others read all deliverables to check understanding
  - if not completed during meeting or team member not at meeting, email assignment after meeting to everybody; no objection within 24h counts as agreement with task assignment

COMMON SOURCES OF CONFLICT

- Different team members have different working patterns and communication preferences
    - e.g., start early vs close to deadline
    - e.g., plan ahead vs try and error
    - e.g., react to every notification vs reduce distractions and read email once a day
    - *discuss and set explicit expectations; talk about conflicts*
- Different abilities, unexpected difficulties
    - work in pairs, plan time for rework and integration
    - replan, contribute to teams in different ways
    - *work around it, it's the team's responsibility*
- Unreliable team members, poor team citizenship
    - e.g., not starting the work in agreed time, not responding, not attending meetings
    - have written clear deliverables with deadlines
    - *talk about it within team, talk to course staff, peer grading*

MEETING TIPS

- Regular 1h meeting, assign moderator who keeps time
- Longer work/integration meeting with needed team members as needed
- If using zoom: Use video, muting often not needed in small groups
- Use Slack/chat deliberately
    - consider chat ephemeral, don't expect everybody to catch up on all old messages
    - explicitly tag people if you need their input
    - separate social communication from work communication, urgent from not urgent
    - discuss non-urgent, long-term things outside of chat associated with topic (issue tracker, Google doc, ...)
- Reserve time to reflect on teamwork and discuss possible improvements on communication and process
- Reserve time for socializing and celebrating success

EXPLORE COLLABORATIVE TOOLS

Examples:

- GitHub issue tracker: async topical discussions and todo list
- Google docs: collaborative report editing
- GitHub wiki or markdown files: design and documentation
- AWS Cloud9, Google Colab, and similar: online IDE and pair programming
- Slack: informal communication and pinging people for immediate questions
- Zoom: planned and ad-hoc meetings

NOW: FIRST TEAM MEETING

- Sending teams into breakout rooms
- Say hi, introduce yourself
    - Name? SE or ML background? Location and time zone? Favorite movie? Fun fact?
    - Find time for first team meeting in next few days (remote/in person?)
    - Agree on primary communication until team meeting
    - Pick a movie-related team name
    - Fill out slide for your team: ...
- No fixed end. Close zoom when done. Feel free to hang out. See you next week.