

Machine Learning in Production Teamwork - A Primer

Now: First Short Team Meeting (10 min)

- Move to table with your team number
- Say hi, introduce yourself: Name? SE or ML background? Favorite movie? Fun fact?
- Find time for first team meeting in next few days
- Agree on primary communication until team meeting
- Pick a movie-related team name, post team name and tag all group members on slack in #social

The Importance of Teamwork

Virtually all software projects are done in teams

ML-enabled projects need to bring together different backgrounds

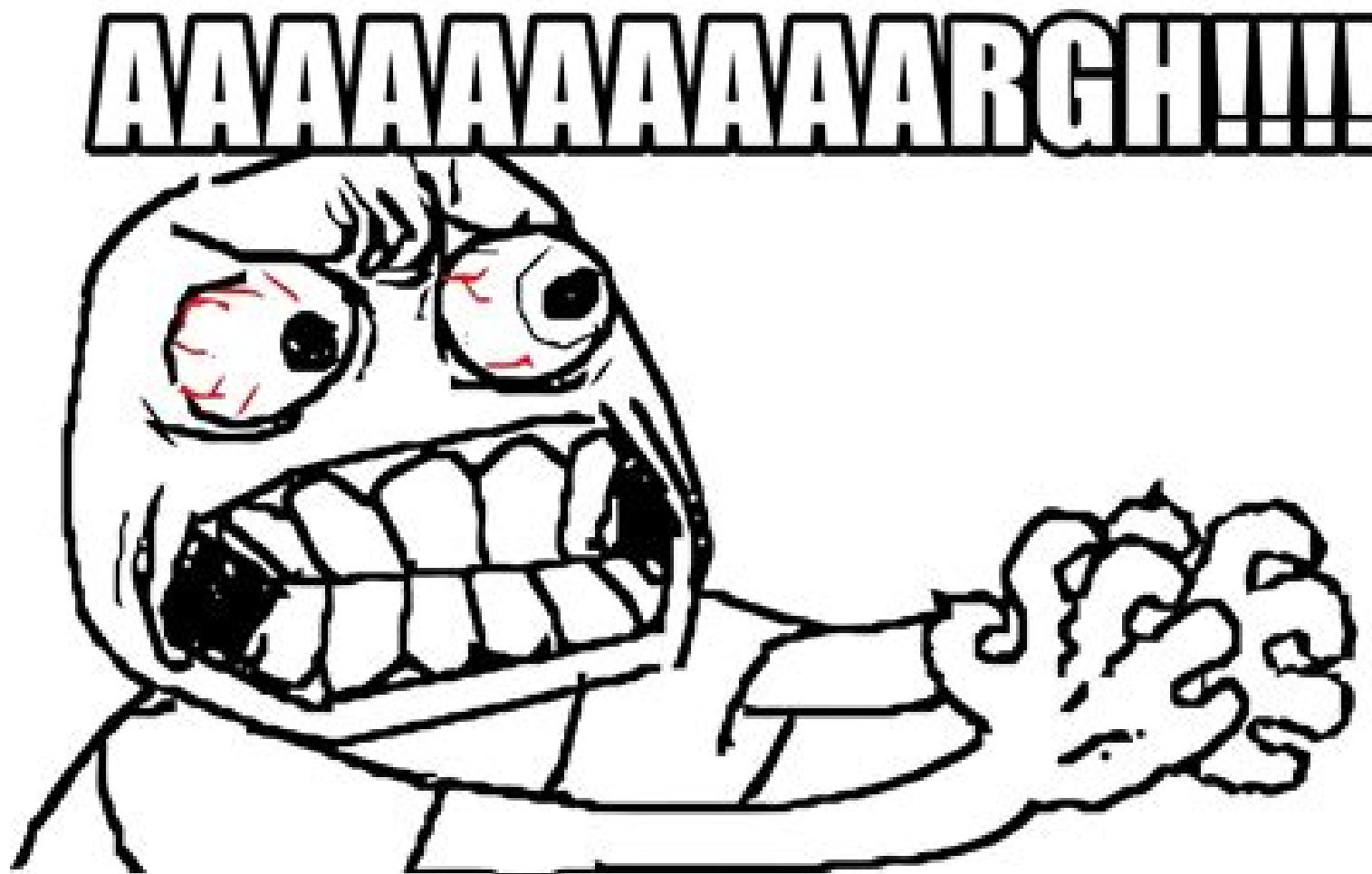
Good teams make it fun to work together

Learn from each other

Limited influence in selecting/firing team members in most organizations

Peer performance evaluations common in industry (e.g. [Google's Process](#))

Who has had bad experiences in teams?
Student teams? Teams in industry?



Some past complaints

- "M. was very pleasant and would contribute while in meetings. Outside of them, he did not complete the work he said he would and did not reach out to provide an update that he was unable to. When asked, on the night the assignment was due, he completed a portion of the task he said he would after I had completed the rest of it."
- "Procrastinated with the work till the last minute - otherwise ok."
- "He is not doing his work on time. And didnt check his own responsibilities. Left work undone for the next time."
- "D. failed to catch the latest 2 meetings. Along the commit history, he merely committed 4 and the 3 earliest commits are some setups. And the latest one commits is to add his name on the meeting log, for which we almost finished when he joined."
- "Unprepared with his deliverables, very unresponsive on WhatsApp recently, and just overall being a bad team player."
- "Consistently failed to meet deadlines. Communication improved over the course of the milestone but needed repeated prompts to get things done. Did not ask for help despite multiple offers."

Common Frustration Themes in Student Teams

No visible progress until last minute

Late work

Incomplete or low quality solutions at integration

Unresponsive team members

Passive, uninterested team members without initiative

Needs lots of reminding and help

Common Sources of Frustrations

- Priority differences ("10-601 is killing me, I need to work on that first", "I have dance class tonight")
- Ambition differences ("a B- is enough for graduating")
- Ability differences ("incompetent" students on teams)
- Working style differences (deadline driven vs planner)
- Communication preferences differences (avoid distraction vs always on)
- In-team competition around grades (outdoing each other, adversarial peer grading)

How would you handle...

One team member has very little technical experience and is struggling with basic Python scripts and the Unix shell. It is faster for other team members to take over the task rather than helping them.

How would you handle...

You divide the work and but when you try to integrate on the evening before the deadline you learn that one team member has failed to complete their part. They tried the day before, but got stuck with a compatibility problem.

How would you handle...

After last minute stress at the last assignment, your team agrees to start earlier and to integrate at a milestone days before the deadline to leave a buffer. Yet you see little progress from half the team in GitHub and hardly anybody responds in Slack. Little is done at the agreed milestone. The work gets done before the deadline, but with the same stress as in the last assignment.

How would you handle...

This homework is low priority for one team member. They rarely contribute beyond the bare minimum at the last minute.

How would you handle...

This homework is low priority for one team member. They rarely contribute beyond the bare minimum at the last minute.

The rest of the team grudgingly compensates and achieves full points for the assignment. You do not feel comfortable criticizing the student as it may negatively affect their grade.

Summary: How would you handle...

1. One team member has very little technical experience and is struggling with basic Python scripts and the Unix shell. It is faster for other team members to take over the task rather than helping them.
2. You divide the work and but when you try to integrate on the evening before the deadline you learn that one team member has failed to complete their part. They tried the day before, but got stuck with a compatibility problem.
3. After last minute stress at the last assignment, your team agrees to start earlier and to integrate at a milestone days before the deadline to leave a buffer. Yet you see little progress from half the team in GitHub and hardly anybody responds in Slack. Little is done at the agreed milestone. The work gets done before the deadline, but with the same stress as in the last assignment.
4. This homework is low priority for one team member. They rarely contribute beyond the bare minimum at the last minute. (The rest of the team grudgingly compensates and achieves full points for the assignment. You do not feel comfortable criticizing the student as it may negatively affect their grade.)

Teamwork Policies and Support in this Course

- Teams can set their own priorities and policies -- do what works for you, experiment
 - Not everybody will contribute equally to every assignment -- that's okay
 - Team members have different strength and weaknesses -- that's good
- We will intervene in *team citizenship* issues!

Golden rule: Try to do what you agreed to do by the time you agreed to. If you cannot, seek help and communicate clearly and early.

Team Citizenship

- Be responsive and responsible
- Come to meetings on time, participate actively
- Stick to commitments, work on assigned tasks
- When problems, reach out, replan, communicate early, be proactive
- (Replanning and dealing with mistakes is normal)

We will adjust grades if complains about:

- Lack of communication
- Disrespectful or dismissive communication
- Not attending team meetings (without excuse)
- Blowing internal deadlines without communication
- Failing to complete agreed tasks without timely communication

Peer Grading Process

- After every milestone you can provide feedback for team citizenship problems in your team
- Rating form and text field, explaining what the issue is
- We discard complains without explanation and those beyond team citizenship (e.g, regarding ability or effort)
- Used to immediately adjust grade, forcing the issue in the team
 - See [form](#) to preview effects
 - Can lead to substantial grade adjustments (-10% to -50% common)
- Depending on severity, we will reach out and work with individuals/team on improvement
- *If entire team agrees, this can be used to adjust grades for intentionally imbalanced contributions too*

Team members:	5							
	Rated by Team member 1	Rated by Team member 2	Rated by Team member 3	Rated by Team member 4	Rated by Team member 5	Indiv. Avg.	Team Avg.	Grade adjustment
Team member 1	Good team citizen (100) ▾	100.0	97.6	+2%				
Team member 2	Needs improvement (90) ▾	Good team citizen (100) ▾	Needs improvement (90) ▾	Good team citizen (100) ▾	Deficient (60) ▾	88.0	97.6	-10%
Team member 3	Good team citizen (100) ▾	100.0	97.6	+2%				
Team member 4	Good team citizen (100) ▾	100.0	97.6	+2%				
Team member 5	Good team citizen (100) ▾	100.0	97.6	+2%				

Avoiding Adversarial Peer Grading

- Peer grading focuses only on team citizenship
- Comments are required and read by instructors
- Instructor may follow up individually or with the group, especially in tricky cases; instructor listens to nuance and appeals, may request/collect evidence
- *Avoid avoiding conflict:* Set high standards and give honest feedback before mounting frustration and spiraling problems

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?
- Set intermediate internal deadline for integration
- Write down expectations!
- Set realistic expectations: All have other classes and distractions; communicate availability openly

Step 2: Maintain Accountability

- Team members have different strength and weaknesses -- that's good
- Make use of individual strength of team members (split, pair up, help, ...)
- Usually somebody will take responsibility for team management tasks, consider rotating
- **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
- We will ask for evidence of this with the first milestone

Recall: Common Sources of Conflict

- Different team members have different working patterns and communication preferences
 - e.g., start early vs close to deadline; plan ahead vs try and error
 - e.g., react to every notification vs reduce distractions, 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
 - 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 selected team members as needed
- 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 comm., 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