Final Project

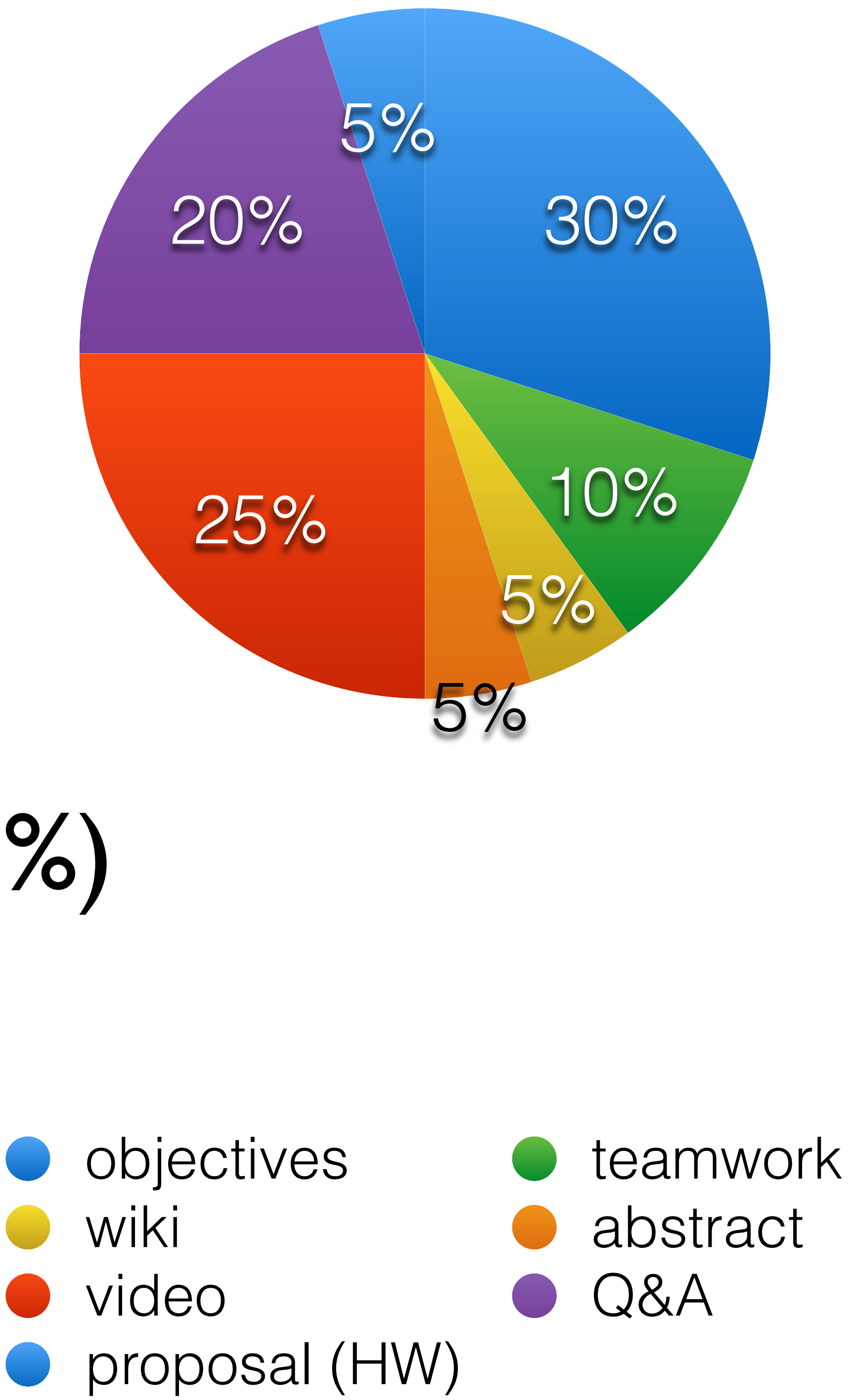
PHY432 Computational Methods in Physics (2025)

Requirements

- team (2 or 3 students)
- GitHub repository
- keep "lab notebook" on wiki page
- final video presentation
  - abstract as PDF
  - description of contributions (PDF)
  - video (slides + narration): mp4
  - individual Q&A (Zoom)

Grading

- individual proposal (HW) (5%)
- code, achieves objectives (30%)
- teamwork (contributions, commits, evidence of communication) (10%)
- keep "lab notebook" on wiki page (5%)
- final video **presentation**
  - abstract (5%)
  - video (25%)
  - individual Q&A (20%)



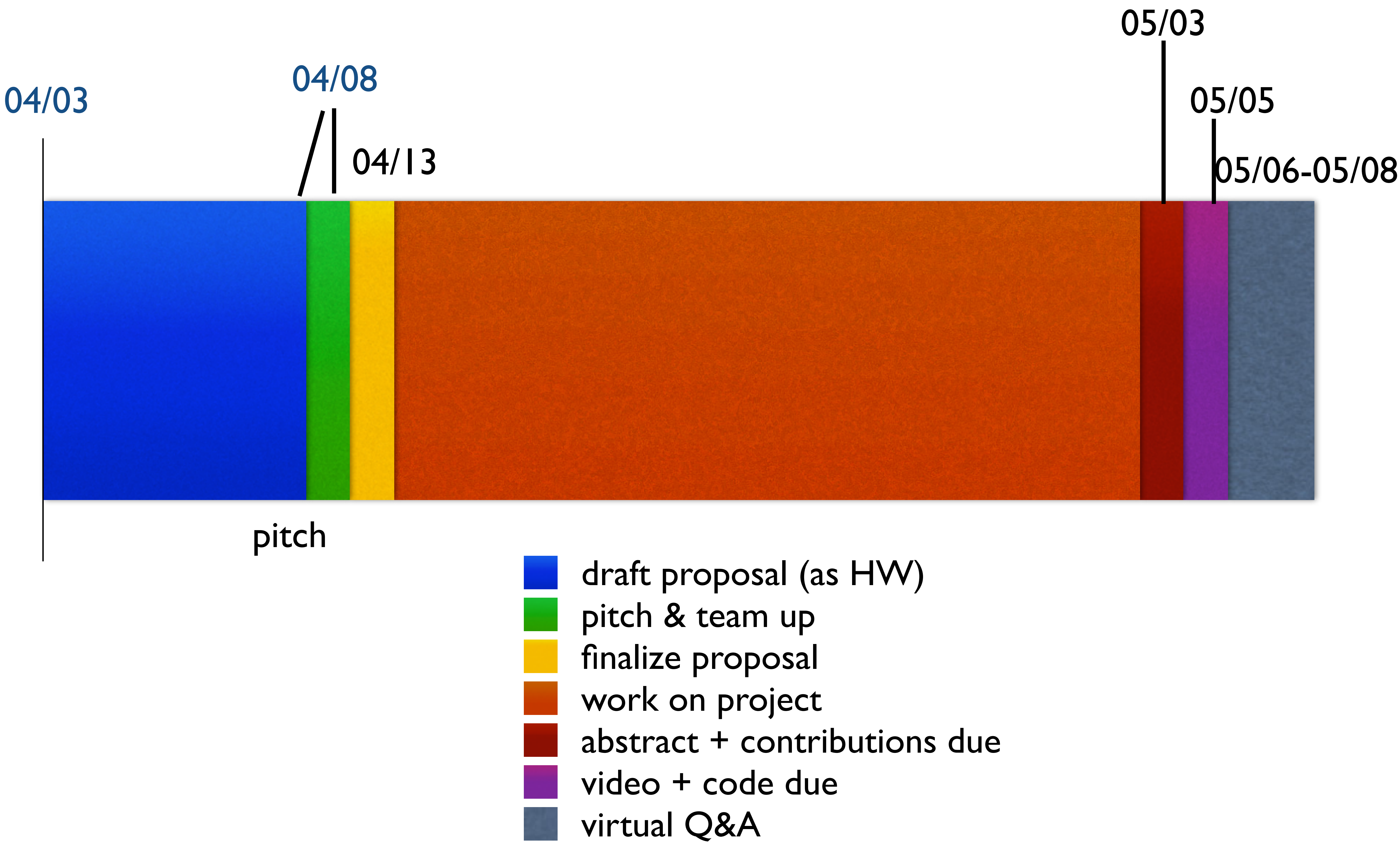
Projects

- Develop one from list (see Appendix A in HW09).
- Propose your own.

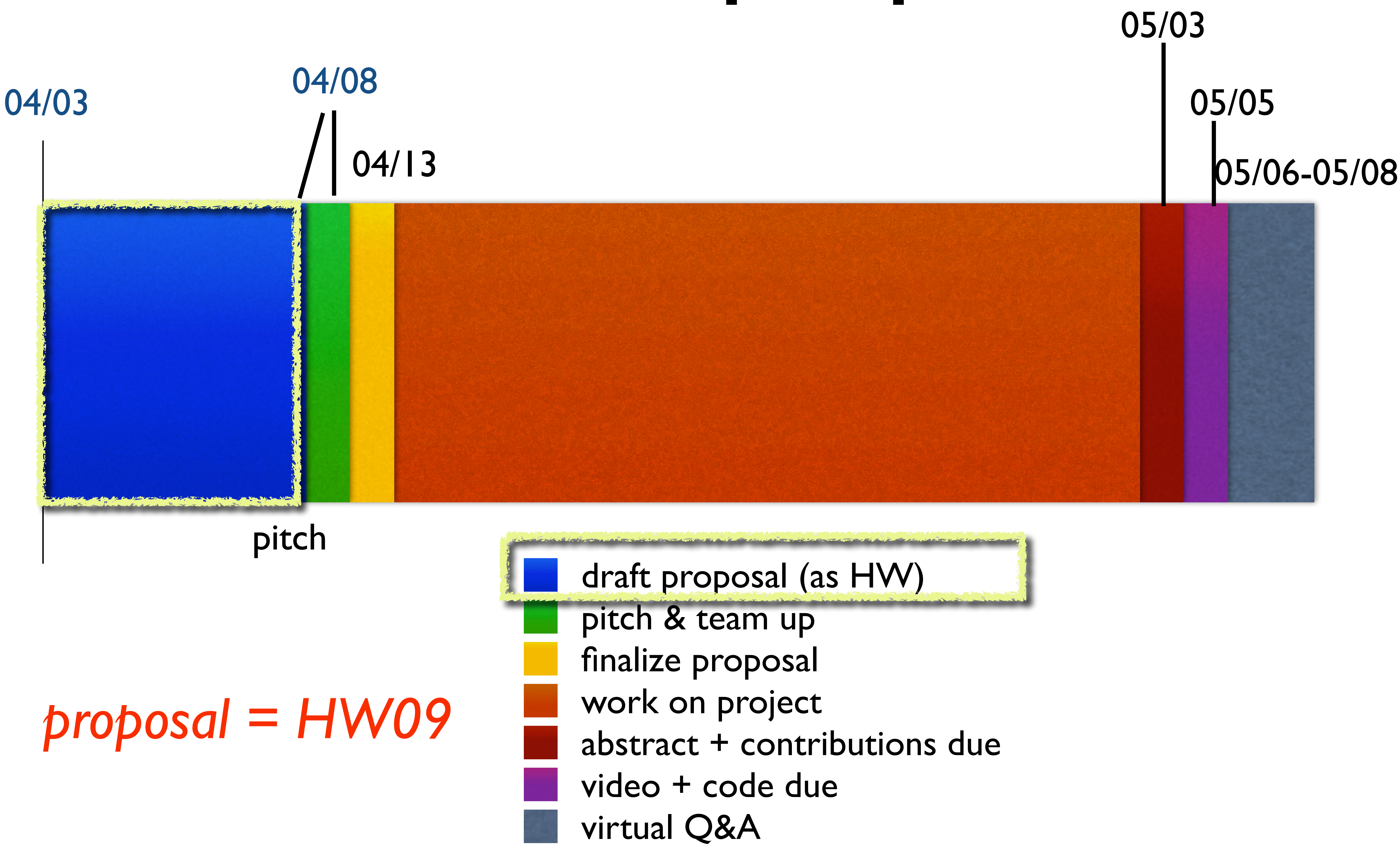
Projects need

- proposal
- pitch

Timeline 2025



Draft proposal



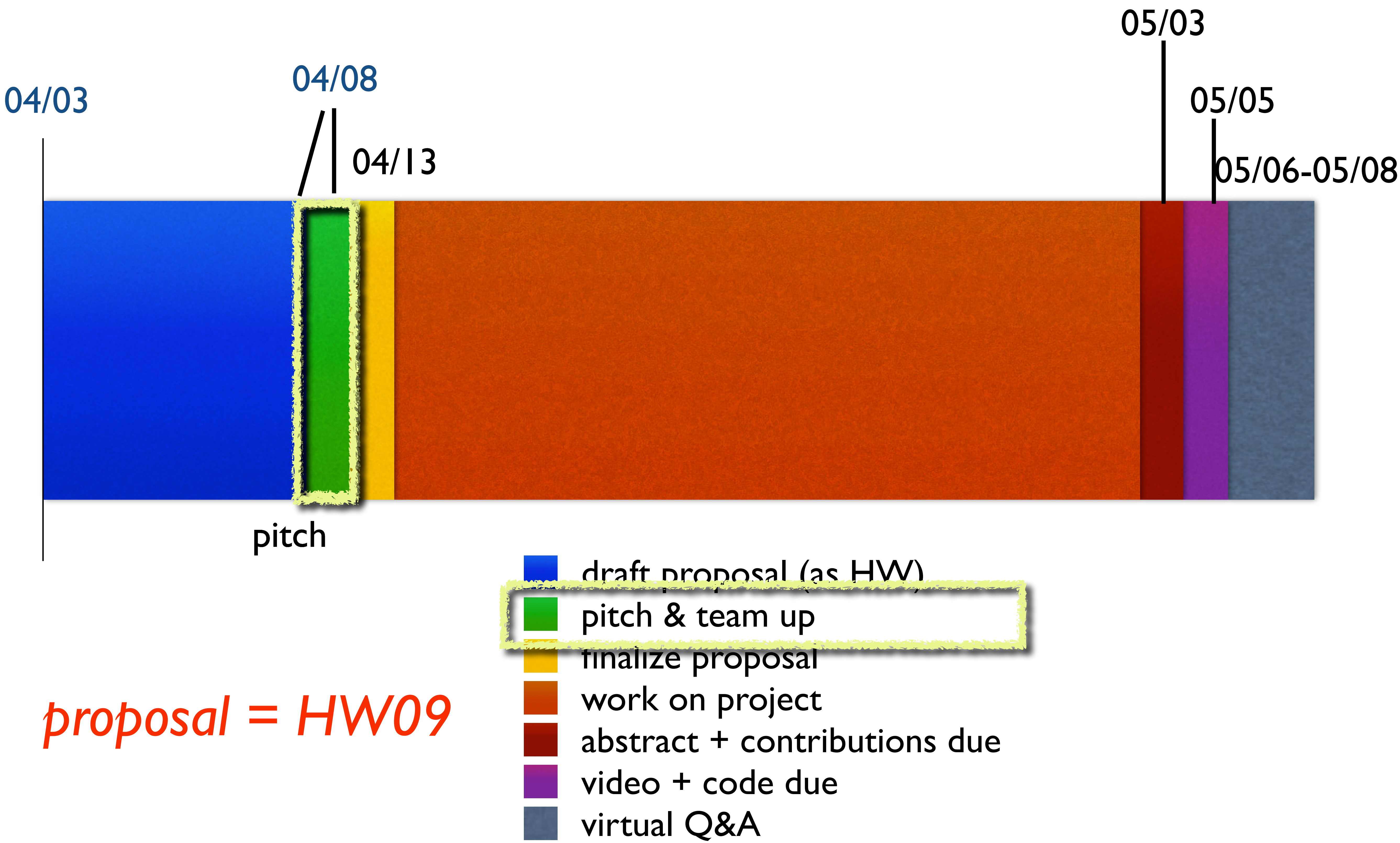
proposal = HW09

Project Proposal

as HW9: due Tue!

- 1 page
- title
- *Problem:* Describe the problem to be solved.
- *Approach:* Algorithms, outline of how you will solve the problem, requirements (e.g. what data needs to be collected?)
- *Objectives*: List 3–5 measurable non-trivial outcomes that you want to achieve; your grade will depend on achieving these objectives

Pitch project & build team

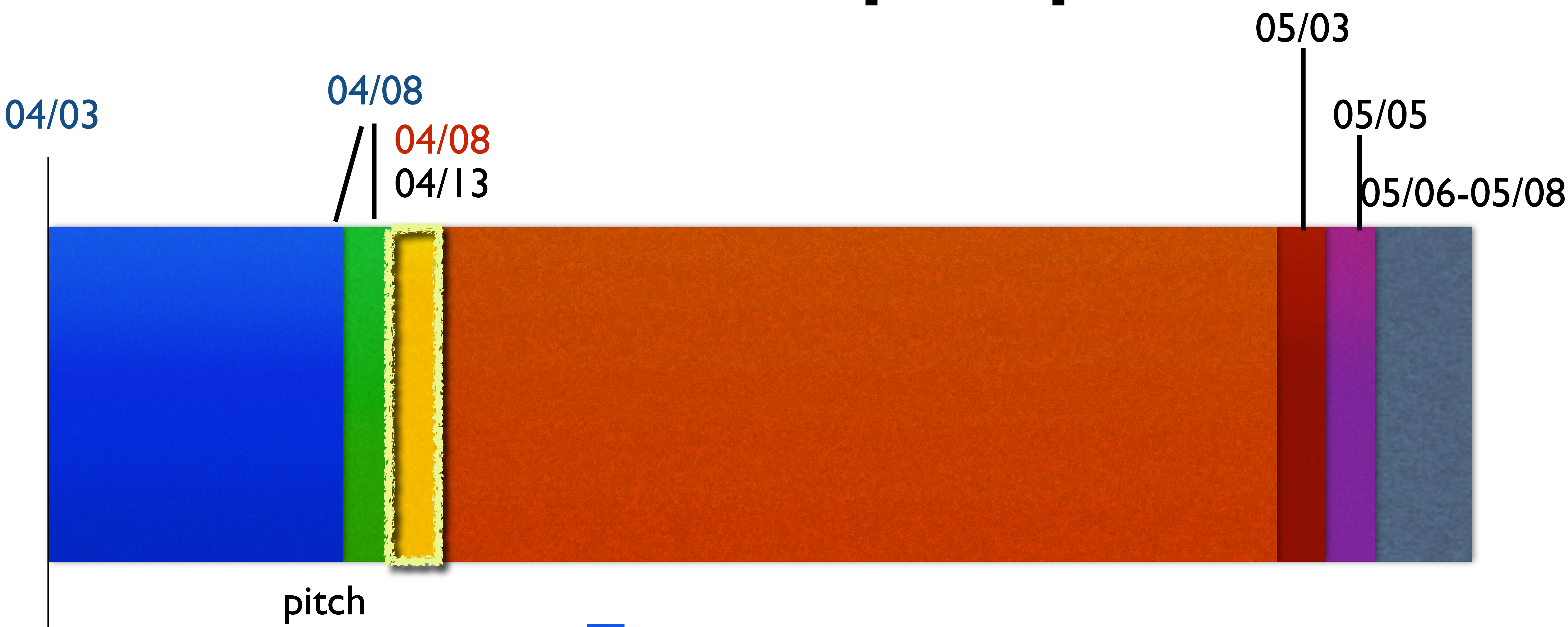


proposal = HW09

Project Pitch

- ≤5 min presentation
- introduce project (*Problem*, *Approach*, *Objectives*)
- attract a team (2 or 3 students)!
- You can only *choose* a project that was pitched… so be prepared to pitch yourself!

Finalize proposal



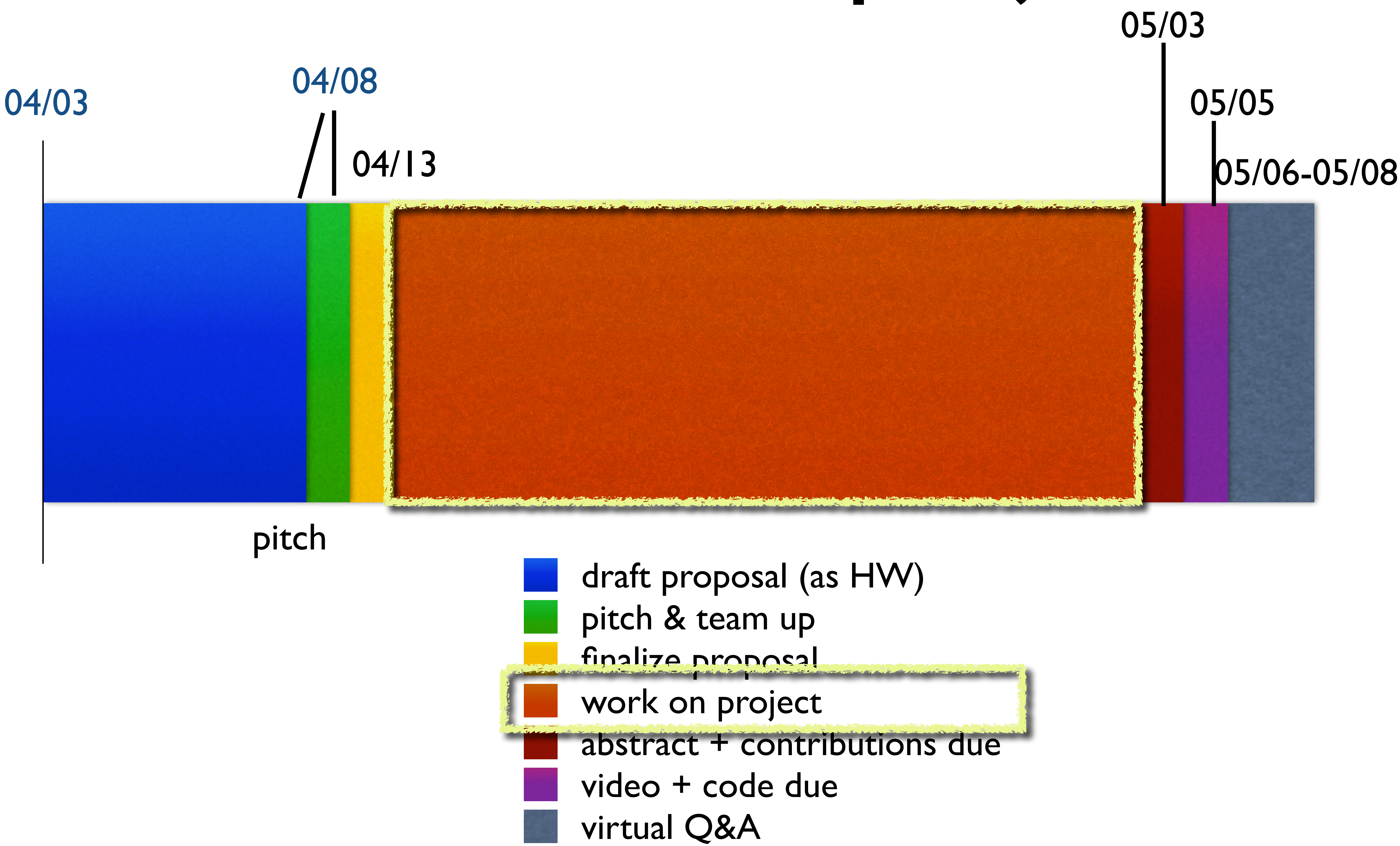
- draft proposal (as HW)
- pitch & team up
- finalize proposal
- work on project
- abstract + contributions due
- video + code due
- virtual Q&A

proposal = HW09

Finalize proposal

1. work as a team to write the final proposal
2. submit to team repository
3. final proposals will be evaluated by instructor and requested changes will be posted as issues: need to be addressed until you get approved ✅

Work on project

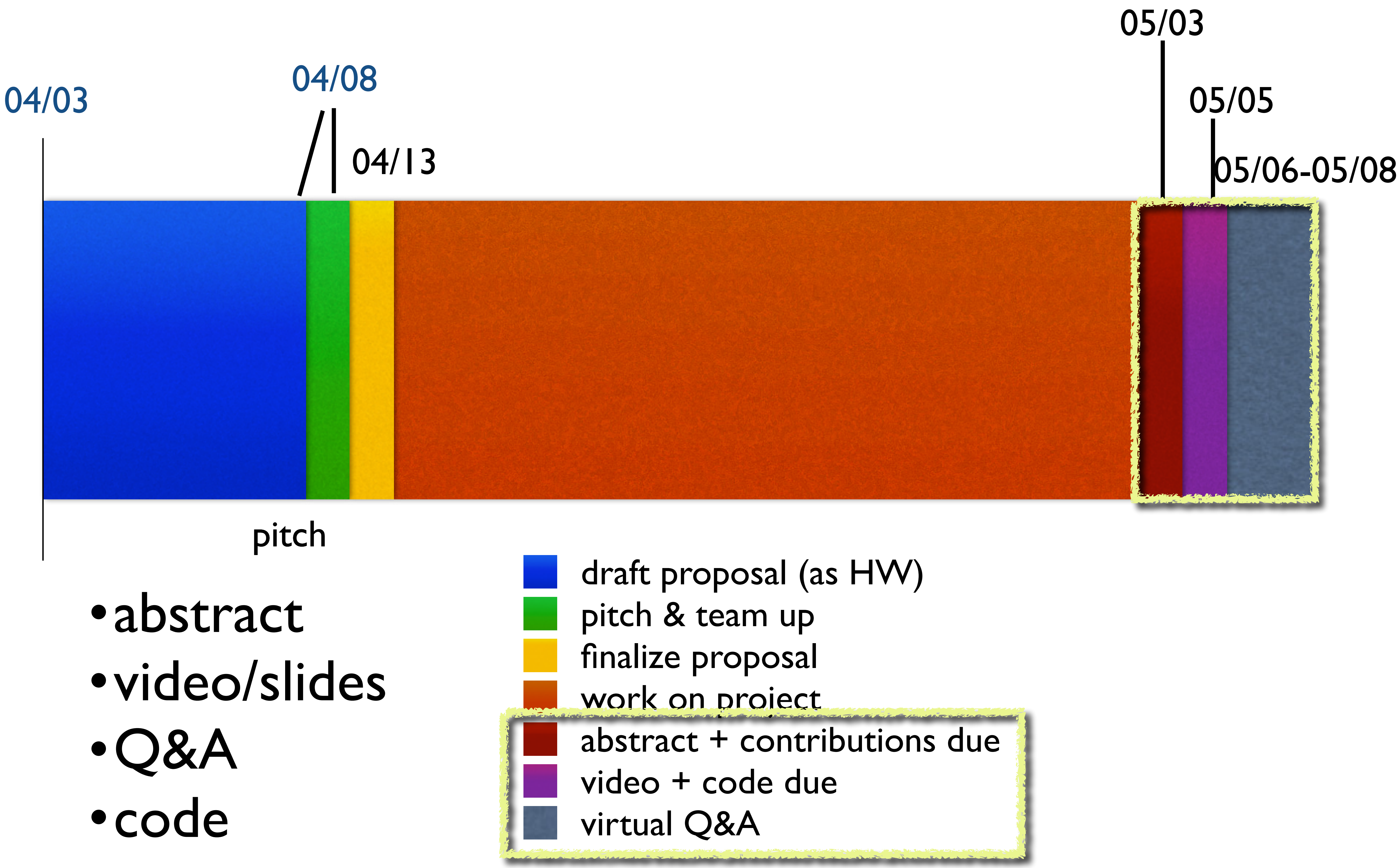


Work on project

- use GitHub **repository** for code
- use GitHub **Wiki** for organization, notes on algorithms, preliminary results
- *optional*: use GitHub issue tracker for assigning tasks and keeping track of bugs and objectives
- Create **abstract** and **video** together.

Start early!

Deliverables



- abstract
- video/slides
- Q&A
- code

Deliverables

1. submit abstract
2. create video presentation (slides, code demo)
3. Virtual Final Symposium (videos will be shared with all groups)
4. individual Q&A via Zoom

More details closer to the end of the project…

Project ideas

- rough outline: expand in your proposal
- see also *Computational Physics* and *Computational modelling and visualisation of physical systems with Python*
- see suggestions in HW9 appendix

1. Monte Carlo Simulation of liquid Argon

- difficulty: 1–2
- implement basic MC for liquid Argon in *NVT* (periodic boundaries, minimum image convention)
  - analyze at different temperatures (phase transition?)
  - calculate equation of state $P(T, \rho)$
- visualize
- extra: implement *NPT* (volume moves)

2. Molecular dynamics of liquid Argon

- difficulty: 1–2
- implement basic MD for liquid Argon in *microcanonical ensemble NVE* (periodic boundaries, minimum image convention)
  - calculate liquid structure (*g(r)*)
  - run for different initial energies, calculate *T* and compare liquid structure
- visualize
- extra: implement *Anderson thermostat* for *NVT*

3. Classical chaotic scattering

- difficulty: 1
- Problem 9.4 in *Computational Physics*
- integrate equations of motions of particles scattering from “4-peak” potential (e.g. with RK4)
- vary parameters
- analyze cross section
- visualize

4. Quantum Mechanics: Wave packet propagation

- difficulty: 3
- See 22.2 and 22.3 in *Computational Physics*
- solve the time-dependent Schrödinger equation for a Gaussian wave packet in different potentials (Visscher or Maestri/Askar&Cakmak algorithm)
- 1D: step barrier, harmonic well, square well; 2D: slit
- calculate transmission/reflection, wave velocity, ...
- visualize

5. Fluid mechanics: 2D Navier-Stokes

- difficulty: 2 (3 without *CP*…)
- See Ch 25 in *Computational Physics*
- solve Navier-Stokes for incompressible 2D flow (finite difference with SOR algorithm)
- velocity (flow) field and vorticity for submerged beam, cylinder, drop shape
- visualize
- vary parameters (e.g. Reynolds number, velocities, boundary conditions)

6. Solar system

- difficulty: 1–2
- simulate the solar system (planets + sun + various comets such as Halley's comet) using classical Newtonian mechanics
- obtain realistic parameters from NASA
- stability over time?
- effect of planets on comets?
- Or simulate a fictional system such as "Tatooine" with two suns – do stable orbits exist where one would see a two-sun rise?

7. Stock market forecasting with wavelets

- difficulty: 4
- See e.g. Yousefi et al (2005), doi:10.1016/j.chaos.2004.11.015
- implement a wavelet analysis (Daubechies wavelets) for stock market and commodity prices timeseries
- test how well you can forecast (correlation between "predicted" data and data not used for the wavelet analysis)
- vary forecast range

8. Bacterial chemotaxis in 2d and 3d

- difficulty: 2-3 (?)
- Bacteria sense with “run-and-tumble”: switch between straight swimming and random reorientation depending on temporal changes in concentration.
- Build a simple model of a bacteria and investigate under which conditions it can follow a concentration gradient.
- Analyze the average behavior in 2d and 3d.
- Visualize.

9. Analysis of natural motion

- difficulty: 3 (?)
- motion (walking, swimming, flying) can be decomposed into very few components using SVD
- example:
  - Girdhar K, Gruebele M, Chemla YR (2015) The Behavioral Space of Zebrafish Locomotion and Its Neural Network Analog. PLoS ONE 10(7): e0128668. doi:10.1371/journal.pone.0128668
  - *Mission Impossible: Rogue Nation*: https://www.youtube.com/watch?v=0iZ-nQ4yFn4 starting at 0:51.
- use videos and image processing + SVD
- gait analysis: is how you walk like a finger print?
- What are the most complex motions?