CS 4530 & CS 5500 Software Engineering

Lesson 12.1: Planning Software Projects

Jonathan Bell, John Boyland, Mitch Wand
Khoury College of Computer Sciences

Learning Objectives for this Lesson

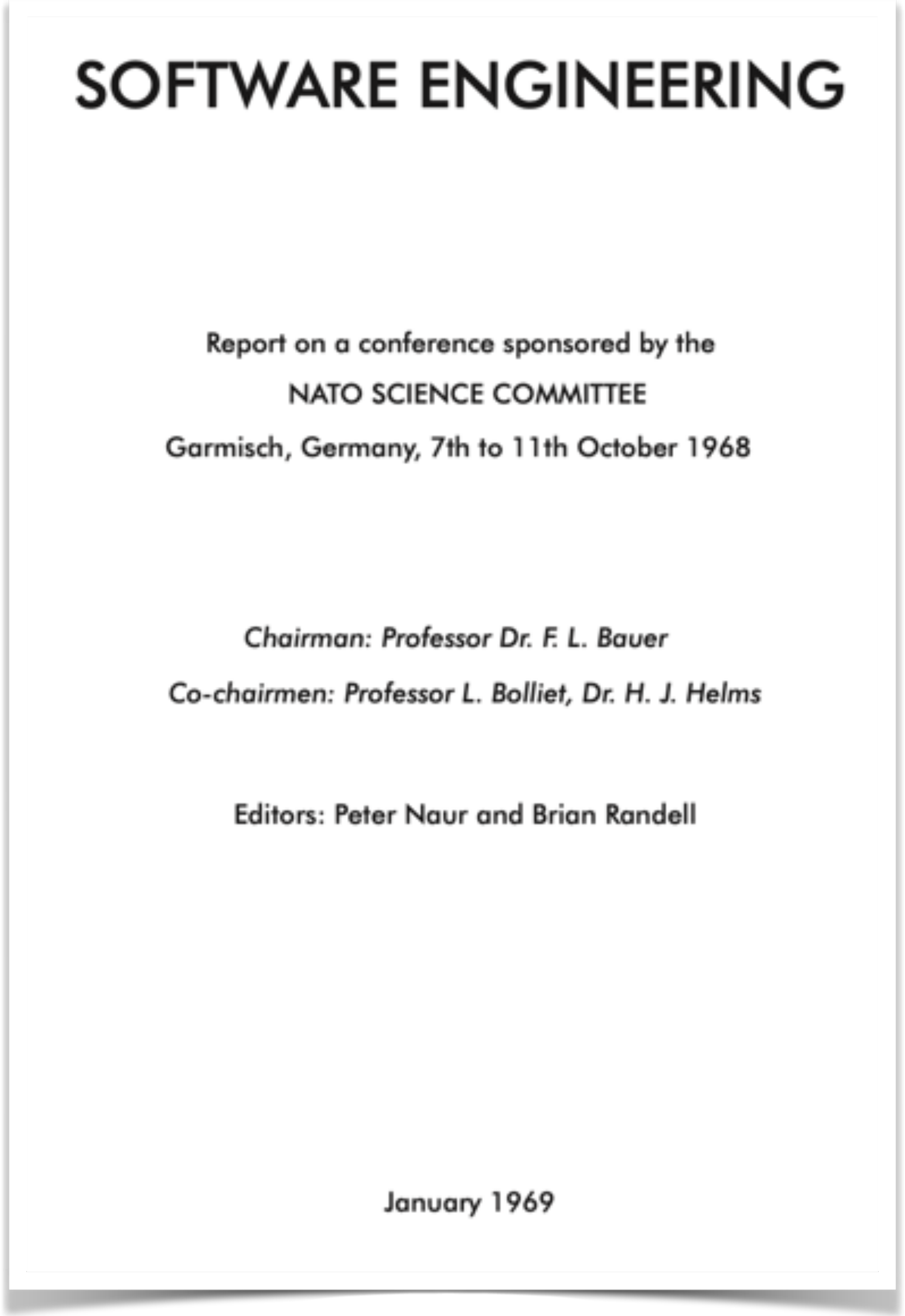
By the end of this lesson, you should be able to…

- Describe how to perform planning and time estimation in an agile model
- Understand why agile processes prefer small team sizes

A brief history of software planning

NATO conference on Software Engineering + Outcomes

- Software was very inefficient
- Software was of low quality
- Software often did not meet requirements
- Projects were unmanageable and code difficult to maintain
- Software was never delivered



SOFTWARE ENGINEERING

Report on a conference sponsored by the
NATO SCIENCE COMMITTEE
Garmisch, Germany, 7th to 11th October 1968

Chairman: Professor Dr. F. L. Bauer
Co-chairmen: Professor L. Bolliet, Dr. H. J. Helms

Editors: Peter Naur and Brian Randell

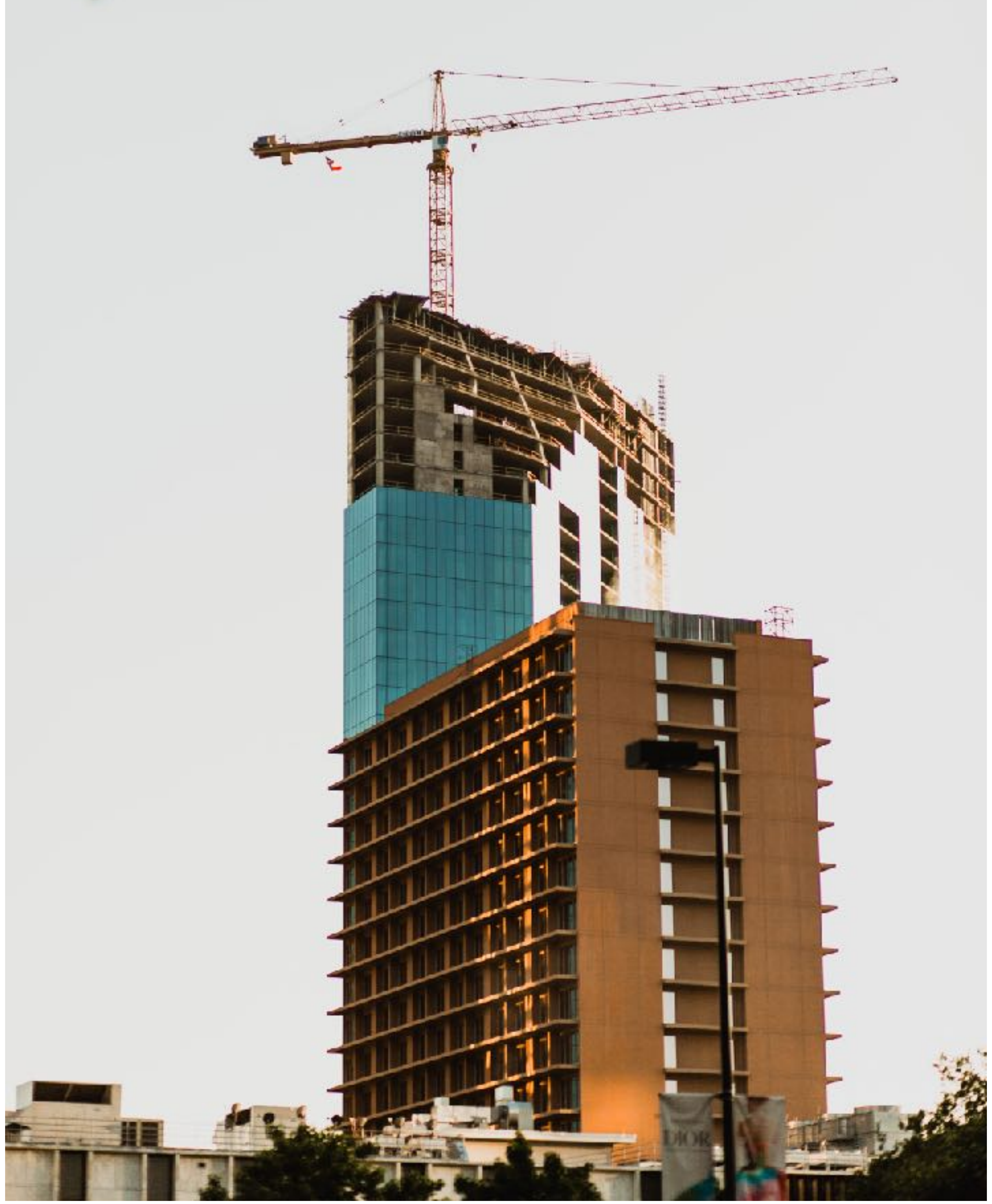
January 1969

A call to action:
We must study
*how to build
software*

Planning Engineering Projects

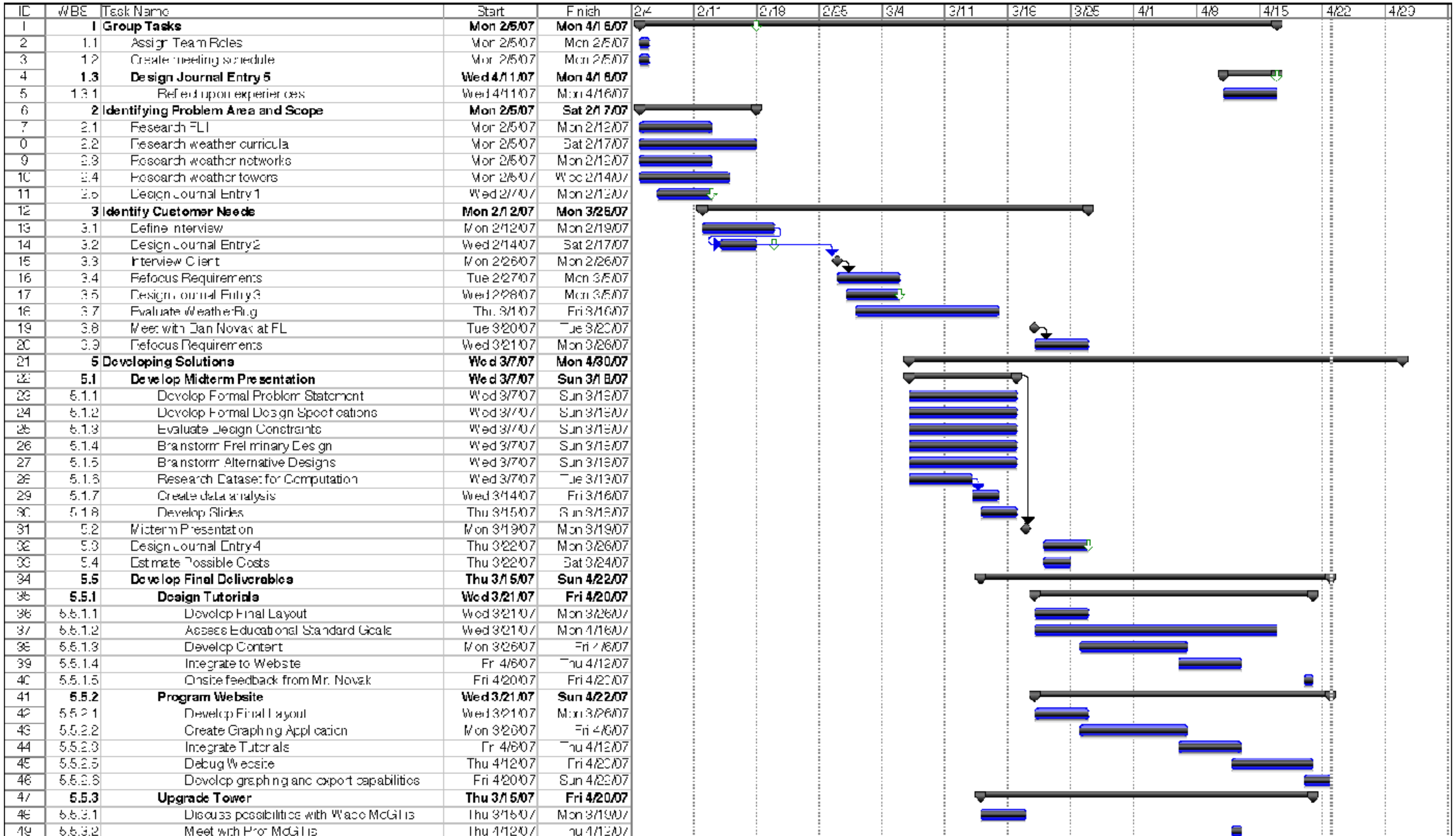
In contrast to software projects:

- Mechanical in nature
- Highly standardized:
  - Design process
  - Materials
  - Construction process



A brief history of software planning

Plan & Document - Gantt Charts



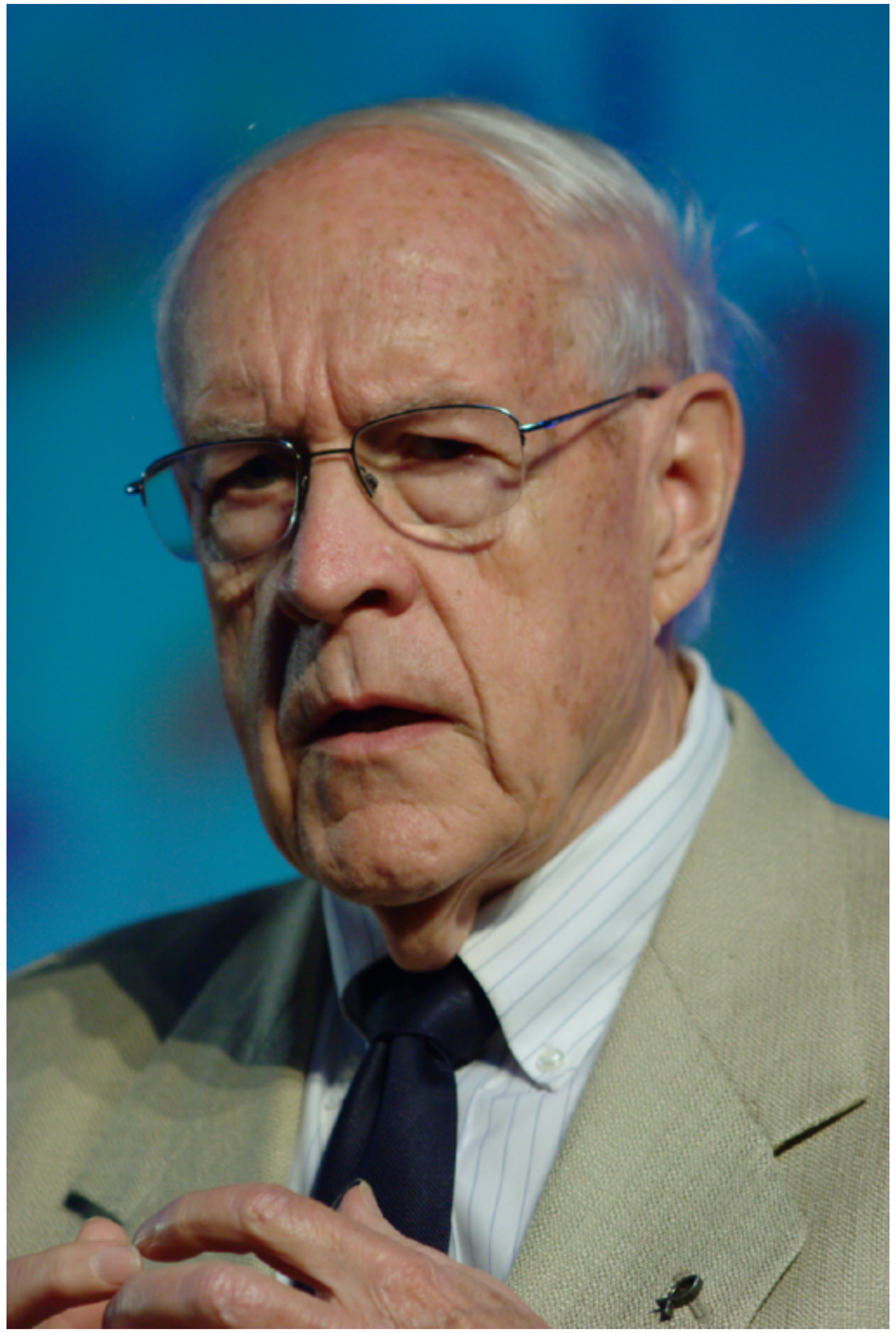
ID	WBS	Task Name	Start	Finish
1	**1**	**Group Tasks**	**Mon 2/5/07**	**Mon 4/16/07**
2	1.1	Assign Team Roles	Mon 2/5/07	Mon 2/5/07
3	1.2	Create meeting schedule	Mon 2/5/07	Mon 2/5/07
4	**1.3**	**Design Journal Entry 5**	**Wed 4/11/07**	**Mon 4/16/07**
5	1.3.1	Reflect upon experiences	Wed 4/11/07	Mon 4/16/07
6	**2**	**Identifying Problem Area and Scope**	**Mon 2/5/07**	**Sat 2/17/07**
7	2.1	Research FLI	Mon 2/5/07	Mon 2/12/07
8	2.2	Research weather curricula	Mon 2/5/07	Sat 2/17/07
9	2.3	Research weather networks	Mon 2/5/07	Mon 2/12/07
10	2.4	Research weather towers	Mon 2/5/07	Wed 2/14/07
11	2.5	Design Journal Entry 1	Wed 2/7/07	Mon 2/12/07
12	**3**	**Identify Customer Needs**	**Mon 2/12/07**	**Mon 3/26/07**
13	3.1	Define Interview	Mon 2/12/07	Mon 2/19/07
14	3.2	Design Journal Entry 2	Wed 2/14/07	Sat 2/17/07
15	3.3	Interview Client	Mon 2/26/07	Mon 2/26/07
16	3.4	Refocus Requirements	Tue 2/27/07	Mon 3/5/07
17	3.5	Design Journal Entry 3	Wed 2/28/07	Mon 3/5/07
18	3.7	Evaluate WeatherBug	Thu 3/1/07	Fri 3/16/07
19	3.8	Meet with Dan Novak at FL	Tue 3/20/07	Tue 3/20/07
20	3.9	Refocus Requirements	Wed 3/21/07	Mon 3/26/07
21	**5**	**Developing Solutions**	**Wed 3/7/07**	**Mon 4/30/07**
22	**5.1**	**Develop Midterm Presentation**	**Wed 3/7/07**	**Sun 3/18/07**
23	5.1.1	Develop Formal Problem Statement	Wed 3/7/07	Sun 3/18/07
24	5.1.2	Develop Formal Design Specifications	Wed 3/7/07	Sun 3/18/07
25	5.1.3	Evaluate Design Constraints	Wed 3/7/07	Sun 3/18/07
26	5.1.4	Brainstorm Preliminary Design	Wed 3/7/07	Sun 3/18/07
27	5.1.5	Brainstorm Alternative Designs	Wed 3/7/07	Sun 3/18/07
28	5.1.6	Research Dataset for Computation	Wed 3/7/07	Tue 3/13/07
29	5.1.7	Create data analysis	Wed 3/14/07	Fri 3/16/07
30	5.1.8	Develop Slides	Thu 3/15/07	Sun 3/18/07
31	5.2	Midterm Presentation	Mon 3/19/07	Mon 3/19/07
32	5.3	Design Journal Entry 4	Thu 3/22/07	Mon 3/26/07
33	5.4	Estimate Possible Costs	Thu 3/22/07	Sat 3/24/07
34	**5.5**	**Develop Final Deliverables**	**Thu 3/15/07**	**Sun 4/22/07**
35	**5.5.1**	**Design Tutorials**	**Wed 3/21/07**	**Fri 4/20/07**
36	5.5.1.1	Develop Final Layout	Wed 3/21/07	Mon 3/26/07
37	5.5.1.2	Assess Educational Standard Goals	Wed 3/21/07	Mon 4/16/07
38	5.5.1.3	Develop Content	Mon 3/26/07	Fri 4/6/07
39	5.5.1.4	Integrate to Website	Fri 4/6/07	Thu 4/12/07
40	5.5.1.5	Onsite feedback from Mr. Novak	Fri 4/20/07	Fri 4/20/07
41	**5.5.2**	**Program Website**	**Wed 3/21/07**	**Sun 4/22/07**
42	5.5.2.1	Develop Final Layout	Wed 3/21/07	Mon 3/26/07
43	5.5.2.2	Create Graphing Application	Mon 3/26/07	Fri 4/6/07
44	5.5.2.3	Integrate Tutorials	Fri 4/6/07	Thu 4/12/07
45	5.5.2.5	Debug Website	Thu 4/12/07	Fri 4/20/07
46	5.5.2.6	Develop graphing and export capabilities	Fri 4/20/07	Sun 4/22/07
47	**5.5.3**	**Upgrade Tower**	**Thu 3/15/07**	**Fri 4/20/07**
48	5.5.3.1	Discuss possibilities with Waco McGinnis	Thu 3/15/07	Mon 3/19/07
49	5.5.3.2	Meet with Prof McGinnis	Thu 4/12/07	Thu 4/12/07

Plan & Document: Problem 1

How do we scale this? The mythical man-month

“Adding manpower to a late software project makes it later”

Fred Brooks, 1975



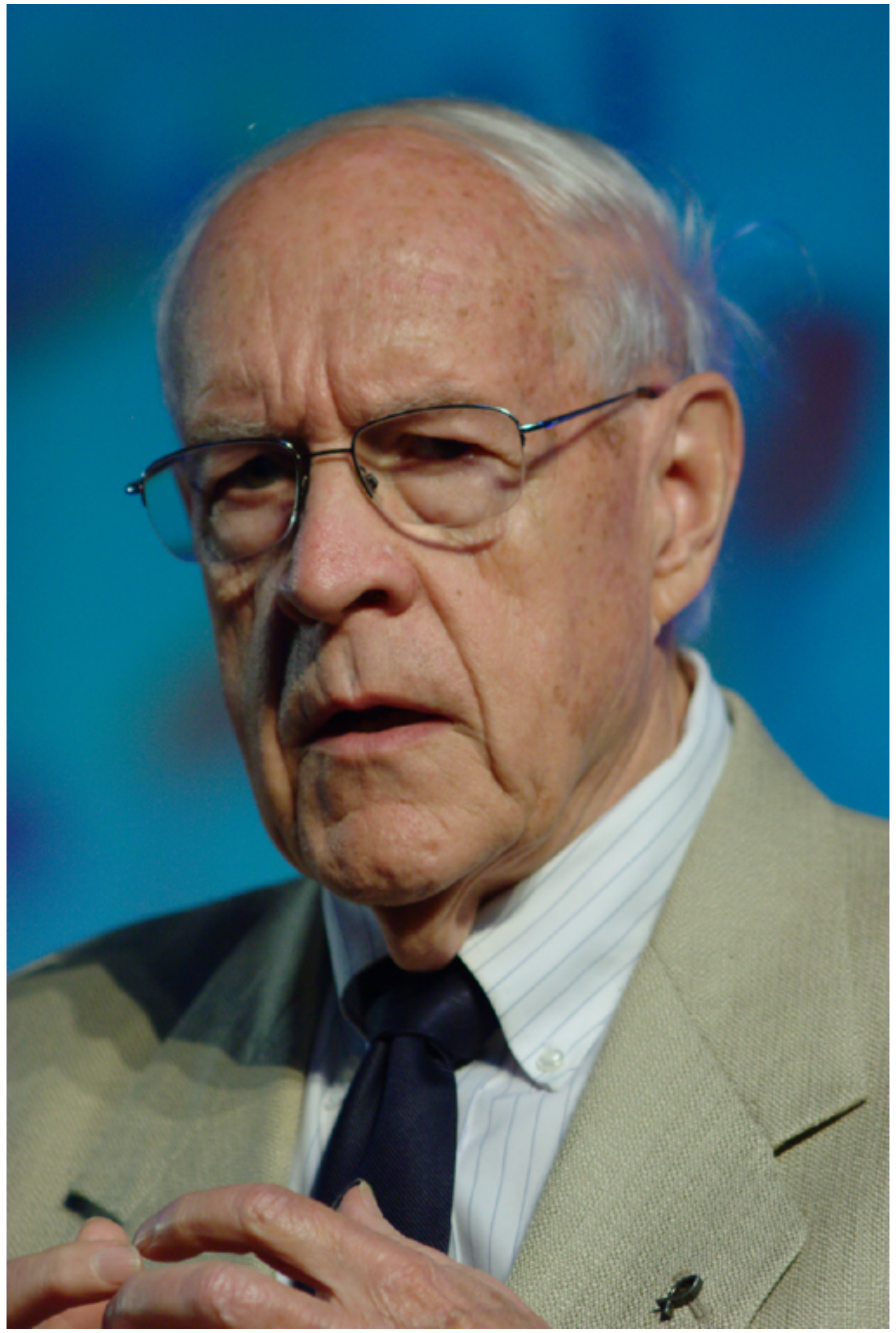
Plan & Document: Problem 2

How long will each task take?

“The management question here, therefore, is not whether to build a pilot system and throw it away. You will do that. The only question is whether to plan in advance to build and throw away, or to promise to deliver the throwaway to customers. Seen this way. the answer is much clearer. Delivering that throwaway to customers buys time, but it does so only at the cost of agony for the user, distraction for the builders while they do the redesign and a bad reputation for the product that the best redesign will find hard to live down.

Hence, plan to throw one away: You will, anyhow.”

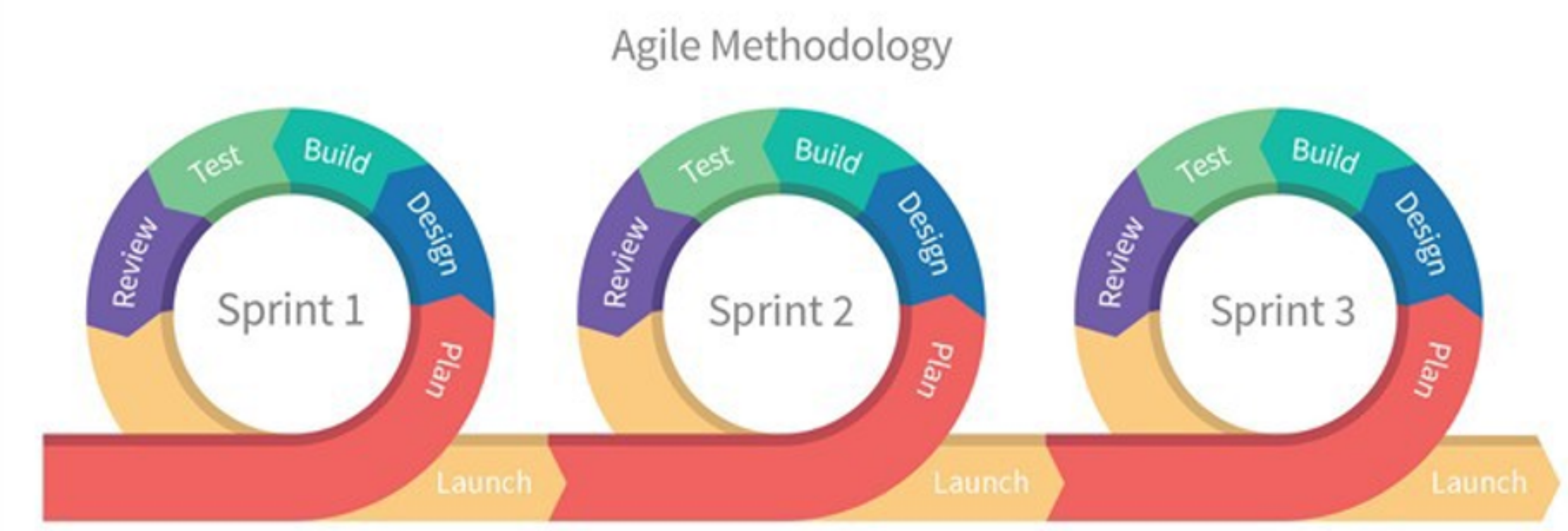
Fred Brooks, 1975



Agile Methodology and Planning

Planning at the per-sprint level

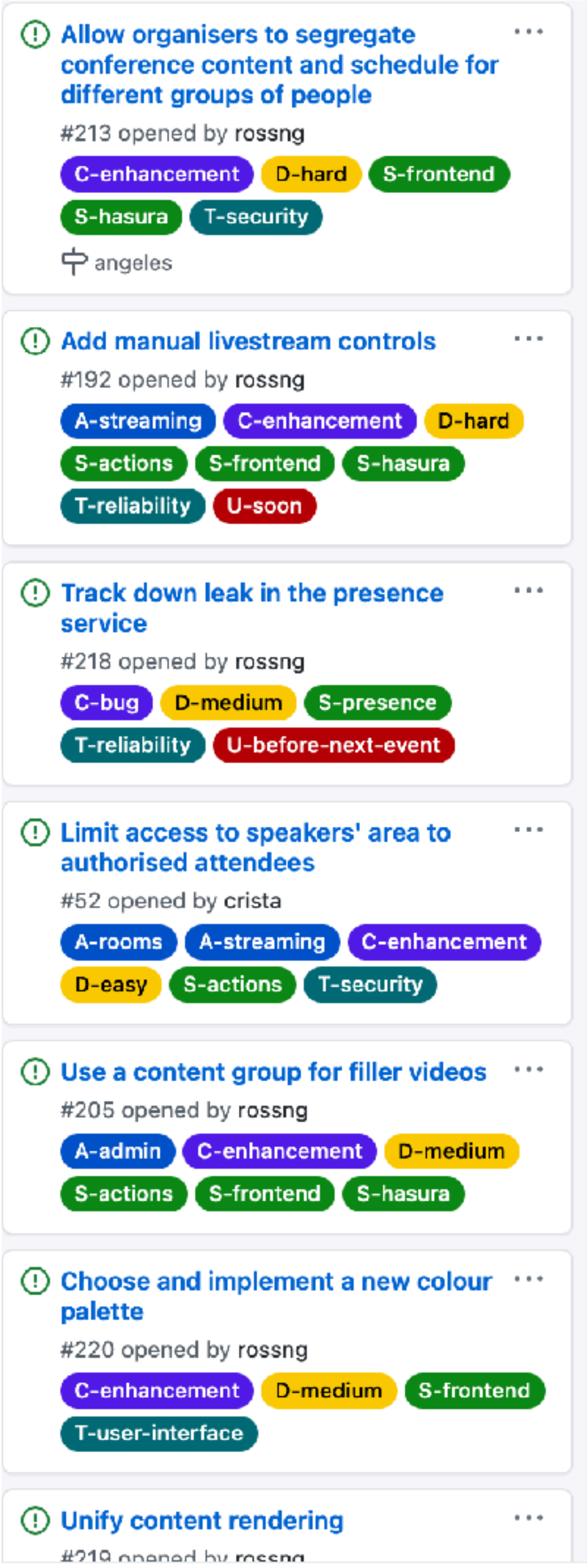
- Break down project into sprints
- Plan each fixed-length sprint independently, fitting the tasks into the sprints
- Key insight: planning might be a guess at first, but gets better with time



Agile Methodology: The Backlog

Organizing development tasks

- Break down big projects into bite-sized chunks
- Estimate the duration of each chunk
- Create a product backlog (has all chunks), sprint backlogs (plan tasks per-sprint)
- Time box sprints: if something doesn't fit, it goes into next sprint
- Focus sprints on business + technical needs



Allow organisers to segregate conference content and schedule for different groups of people

#213 opened by rossng

C-enhancement D-hard S-frontend S-hasura T-security

angeles

Add manual livestream controls

#192 opened by rossng

A-streaming C-enhancement D-hard S-actions S-frontend S-hasura T-reliability U-soon

Track down leak in the presence service

#218 opened by rossng

C-bug D-medium S-presence T-reliability U-before-next-event

Limit access to speakers' area to authorised attendees

#52 opened by crista

A-rooms A-streaming C-enhancement D-easy S-actions T-security

Use a content group for filler videos

#205 opened by rossng

A-admin C-enhancement D-medium S-actions S-frontend S-hasura

Choose and implement a new colour palette

#220 opened by rossng

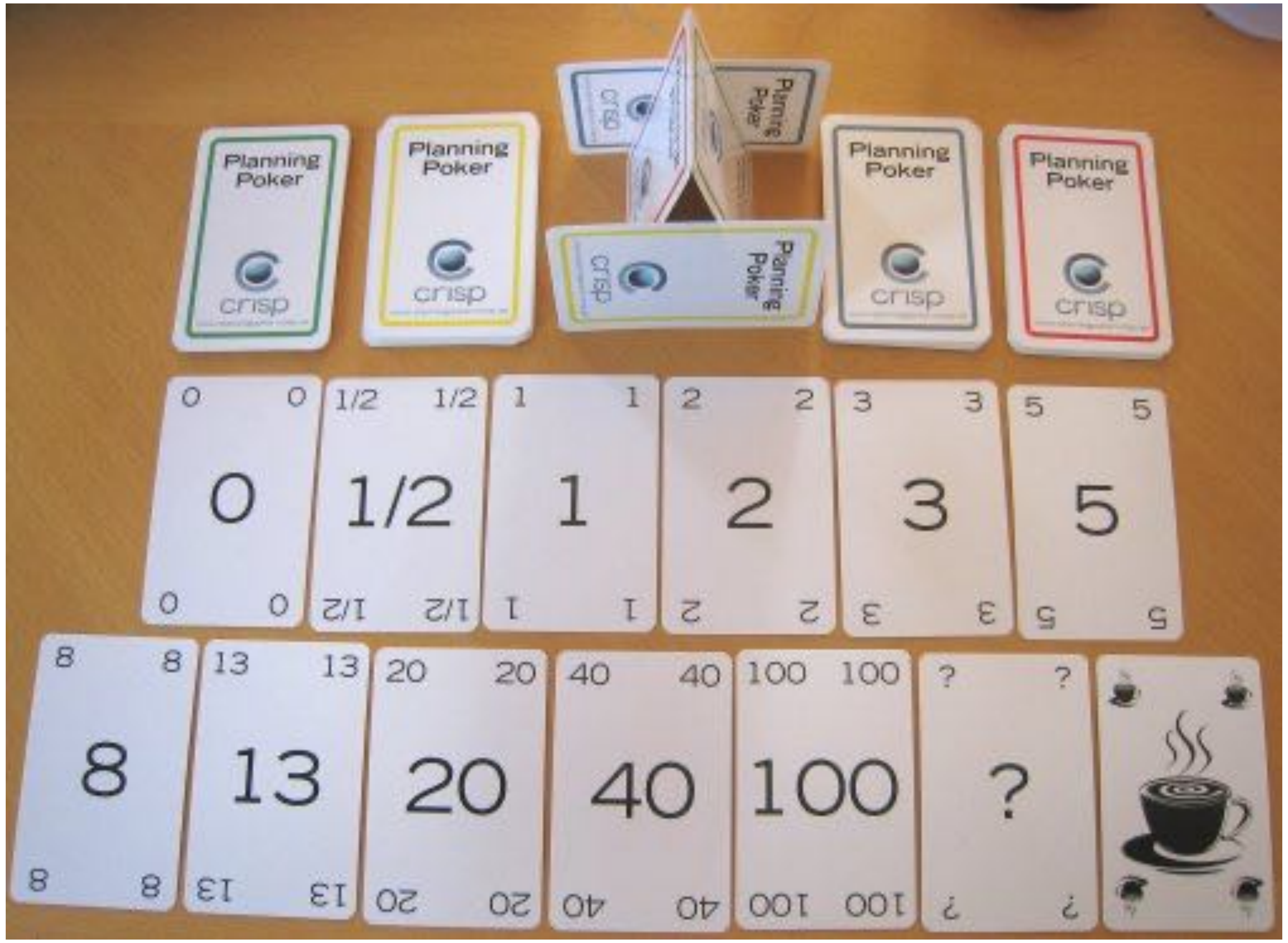
C-enhancement D-medium S-frontend T-user-interface

Unify content rendering

#219 opened by rossng

Agile Methodology: Time Estimation

Planning Poker and Story Points



Agile Methodology: Communications at Scale

How many links are there between people in your team?

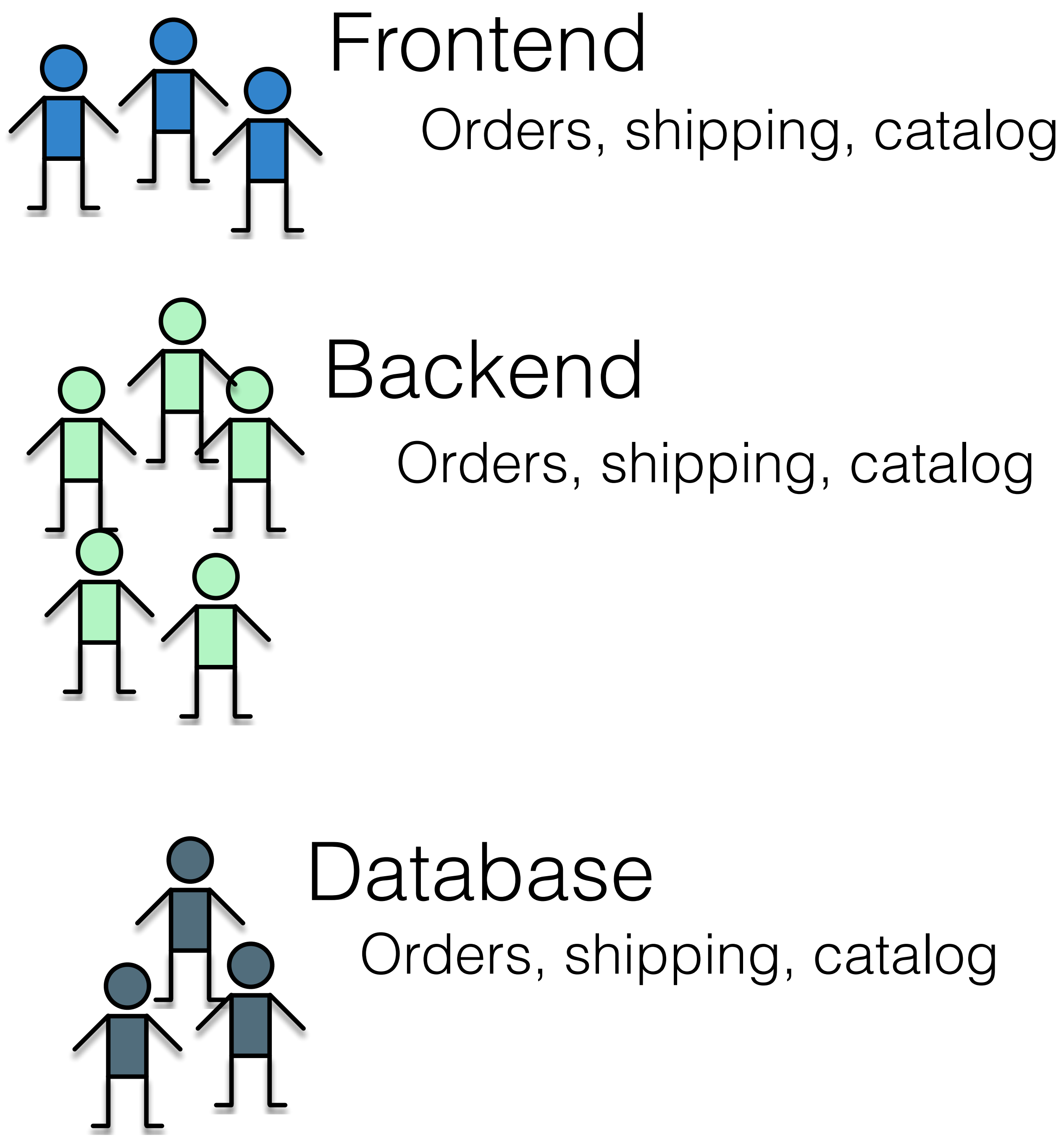
Brooks’ Law: “Adding manpower to a late software project makes it later”

Rationale:

Communication links = $n(n-1)/2$

Agile Methodology: Communications at Scale

Non-agile teams



Frontend

Orders, shipping, catalog

Backend

Orders, shipping, catalog

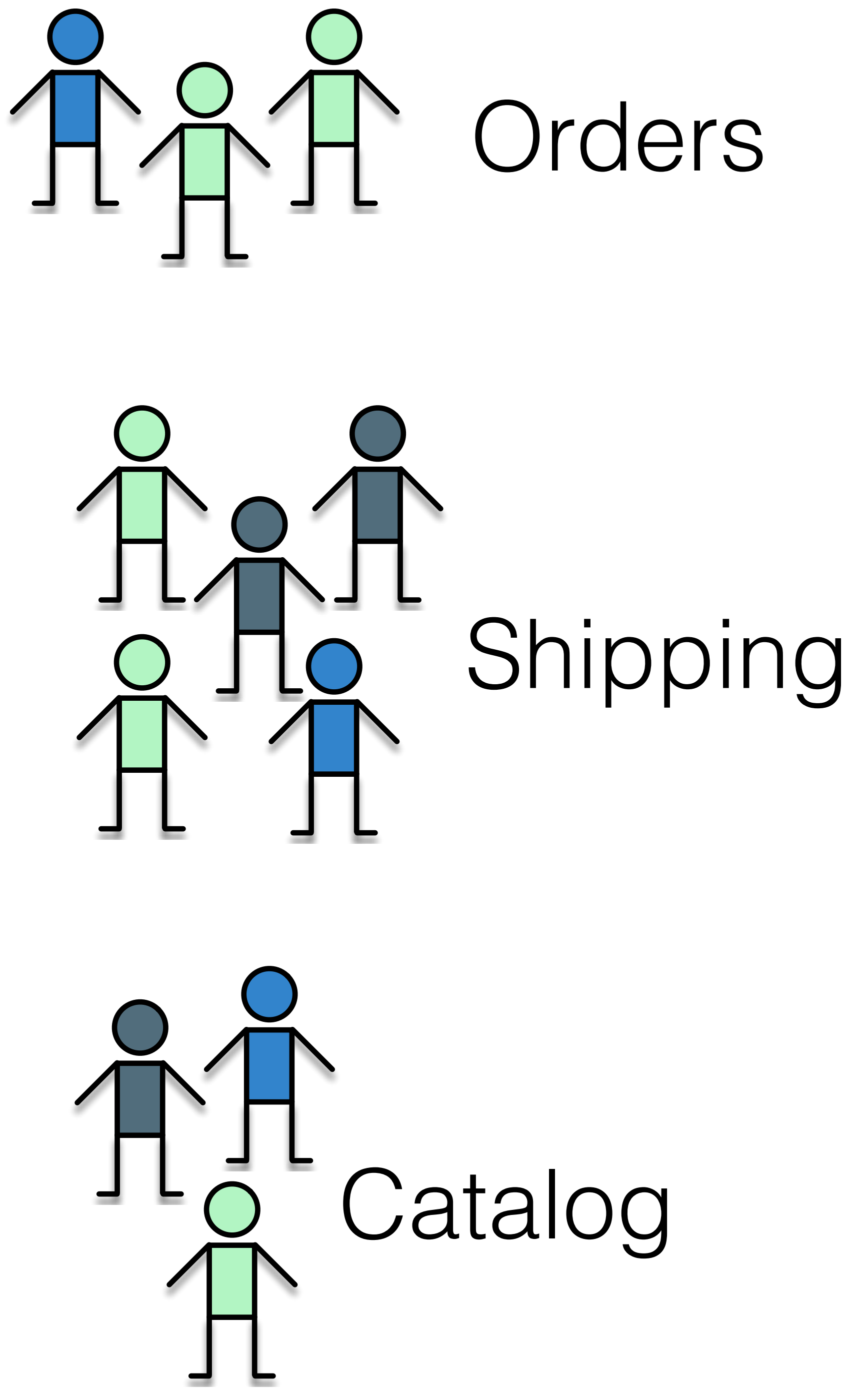
Database

Orders, shipping, catalog

Classic teams:
1 team per "tier"

Agile Methodology: Communications at Scale

Form “Two pizza” teams



Example: Amazon

Teams can focus on one
business task
And be responsible
directly to users

“Full Stack”

“2 pizza teams”

Delivering on a Deadline

This work is licensed under a Creative Commons Attribution-ShareAlike license