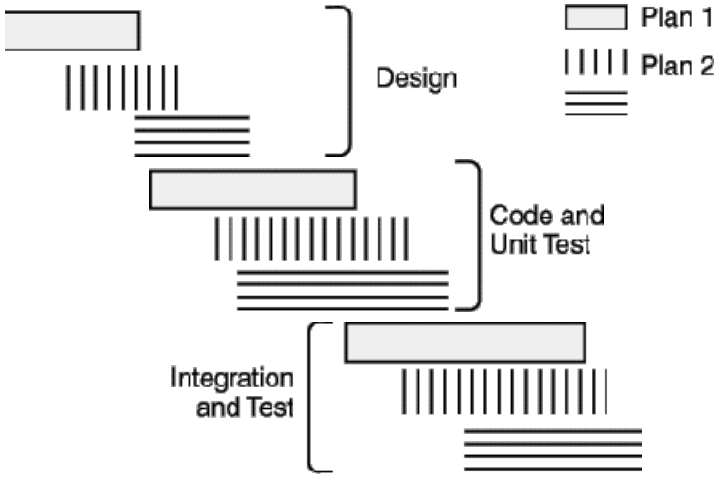
Tracking Project Progress

CSCI 3300/5300

Reading Overview

- Activity Based Measure of Progress: percent of planned activities complete
- Product Based Measure of Progress: percent of completed activities
- Other concepts:
  - Gantt chart
  - Earned Value Analysis

Gantt Chart from your Reading



Questions for Discussion

- What kind of repercussions would a company like this face once they've missed too many deadlines? Is there something that the clients can do beyond hindering the company financially or by damaging their reputation? I feel like the pushing back of deadlines is poor move business wise.
- I have to wonder about how often people can actually meet their expectations. I would think that experienced developers would be able to set reasonable expectations and meet them, but what if the expectations are unattainable? Is it fair to measure progress in a flawed plan?

Statements for Discussion

- Progress management tools have no real application beyond visualizing progress. They are however, an excellent way to hone in on questions that pertain to the overall status. In short, these tools reduce time wasted and clarify the project's situation.
- **Near real time:** the measurements should reflect what is going on in the project now, not what happened a month ago. We want to be managing the present, not the past.
  Prediction: The measure must support projections about future progress. Simply knowing that a project is behind schedule is not enough. We also need to predict when it will be complete.

Clarifications

- I think that having only a manager giving the estimate of how complete a part of a project is one-sided. Having all of the team members evaluate the progress on a task I feel would give a better estimate of the completion of a task.

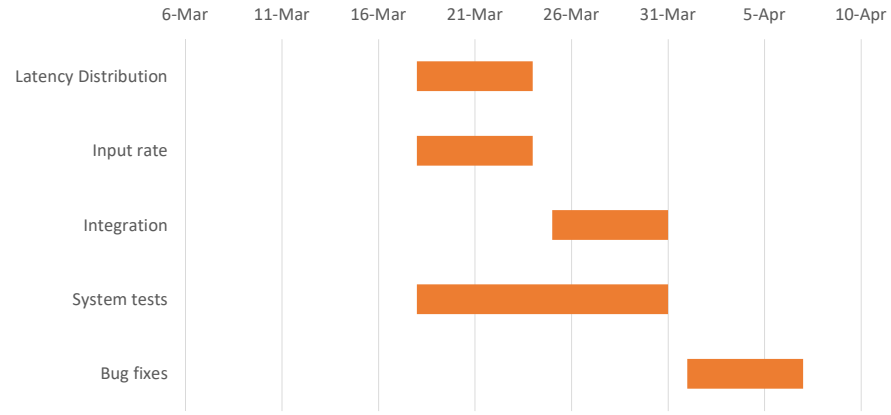
Activity Based Measure of Progress

- Latency distribution calculation: 50% done
  - Update Stats module
  - Unit test
- Input data rates on per-microsecond basis: 50% done
  - Update Input module
  - Unit test
- Integration: 0% done
  - Integrate updated Input module
  - Integrate updated Stats module
- Write system level tests: 30% done
  - Test case 1
  - Test case 2
  - Test case 3
- Execute tests/fix bugs: 0% done

High Level Schedule

	Start	End
Latency Distribution	18-Mar	24-Mar
Input rate	18-Mar	24-Mar
Integration	25-Mar	31-Mar
System tests	18-Mar	31-Mar
Bug fixes	1-Apr	7-Apr

Gantt Chart: expected schedule



Relative weights

weight = duration/Total

	Start	End	Duration	weight
Latency Distribution	18-Mar	24-Mar	6	0.162162
Input rate	18-Mar	24-Mar	6	0.162162
Integration	25-Mar	31-Mar	6	0.162162
System tests	18-Mar	31-Mar	13	0.351351
Bug fixes	1-Apr	7-Apr	6	0.162162
Total			37	1

Total weight must be 1 (100%)

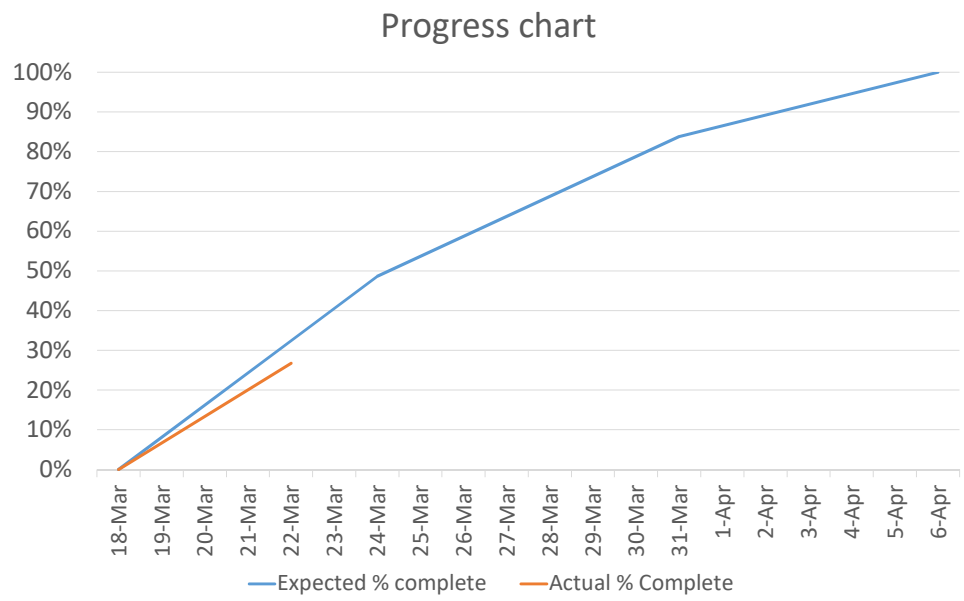
Expected Weighed Percent Complete

	Start	End	Duration	Weight	22-Mar	24-Mar	31-Mar	7-Apr
Latency Distribution	18-Mar	24-Mar	6	0.162	10.81%	16.22%	16.22%	16.22%
Input rate	18-Mar	24-Mar	6	0.162	10.81%	16.22%	16.22%	16.22%
Integration	25-Mar	31-Mar	6	0.162	0.00%	0.00%	16.22%	16.22%
System tests	18-Mar	31-Mar	13	0.351	10.81%	16.22%	35.14%	35.14%
Bug fixes	1-Apr	7-Apr	6	0.162	0.00%	0.00%	0.00%	16.22%
Total			37		32.43%	48.65%	83.78%	100.00%

If a given DATE is after activity's start date and before end date,
expected weighted % complete for that activity for that DATE is:
WEIGHT * (DATE-START)/DURATION * 100%
If DATE is at or after end date,
expected weighted % complete for that activity for that DATE is
WEIGHT * 100%

Actual Weighted Percent Complete per Activity

	weight	% Complete	Weighted % Complete
Latency Distribution	0.162162	50%	8%
Input rate	0.162162	50%	8%
Integration	0.162162	0%	0%
System tests	0.351351	30%	11%
Bug fixes	0.162162	0%	0%
Total	1		27%

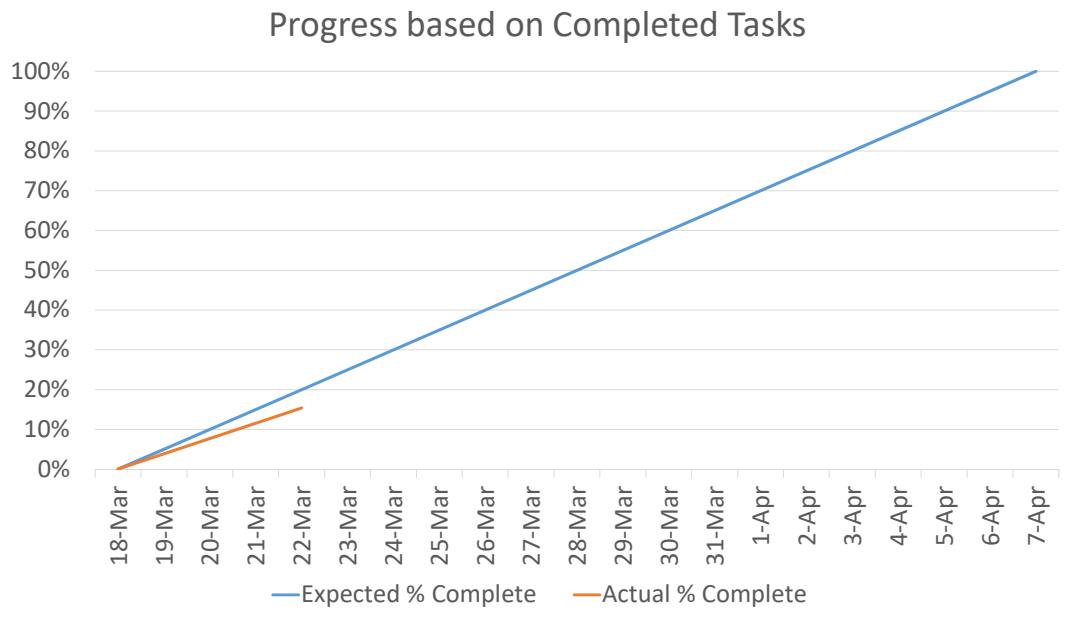


Product Based Measure of Progress

- Only completed items count as Done (no partial credit)
- Tasks are assigned point values (1, 1, 2, 3, 5, 8, 13, …) based on their relative size
- Track completed points versus total points
- Finer grain break down of tasks is beneficial for showing progress

Task Board

Task	Status	Points
Update Stats Module	Done	2
Unit tests for Stats Module	Todo	1
Fix errors found with unit tests (stats module)	Todo	1
Update Input Module	Done	2
Unit tests for Input Module	Doing	1
Fix errors found with unit tests (input module)	Todo	1
Integrate Input module	Todo	2
Integrate Stats Module	Todo	2
Write system test case 1	Doing	3
Write system test case 2	Todo	3
Write system test case 3	Todo	3
Execute tests, fix bugs	Todo	5
Total		26



Which progress report is more accurate and why?

Activity Based

Product Based