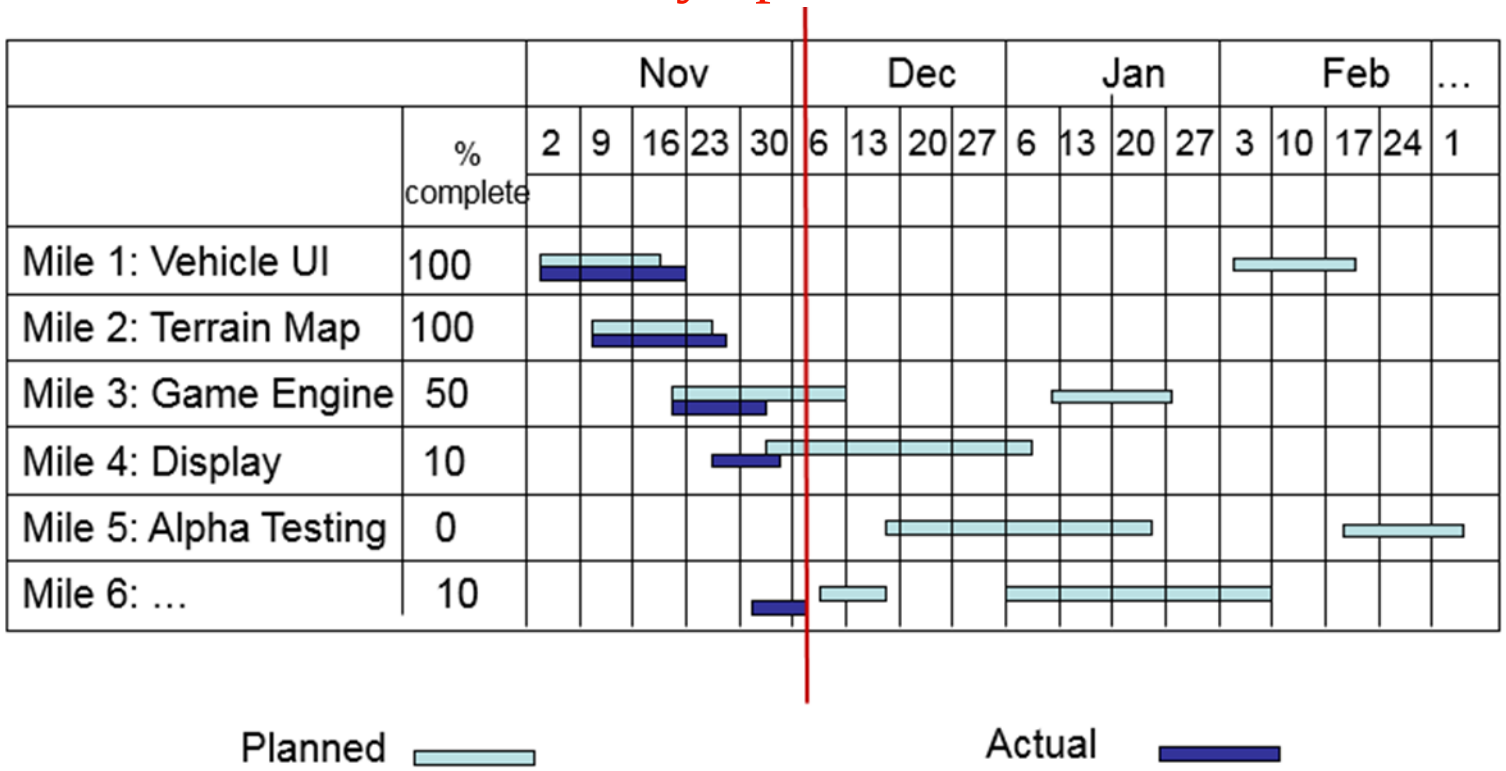
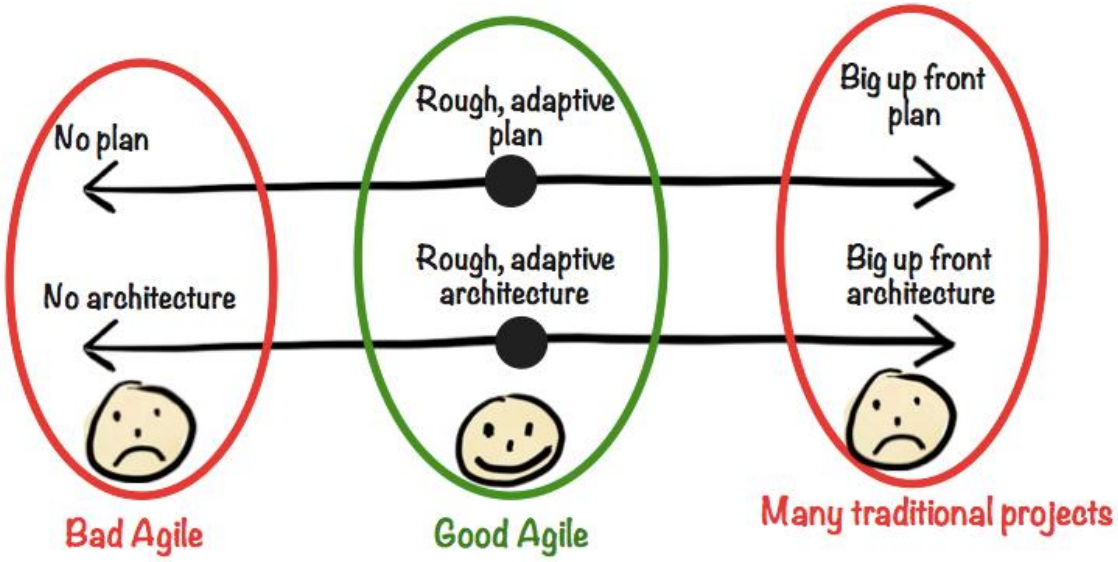
Tracking Capstone Project Progress

- Gantt Charts: Useful for visualizing a project *plan*.
  - However -> much less useful for tracking *progress*.
    - Ex: How much effort is involved with "Vehicle UI" as compared with "Terrain Map" below?
    - Needs to be constantly updated



Revisiting Agile from Project Mgt viewpoint

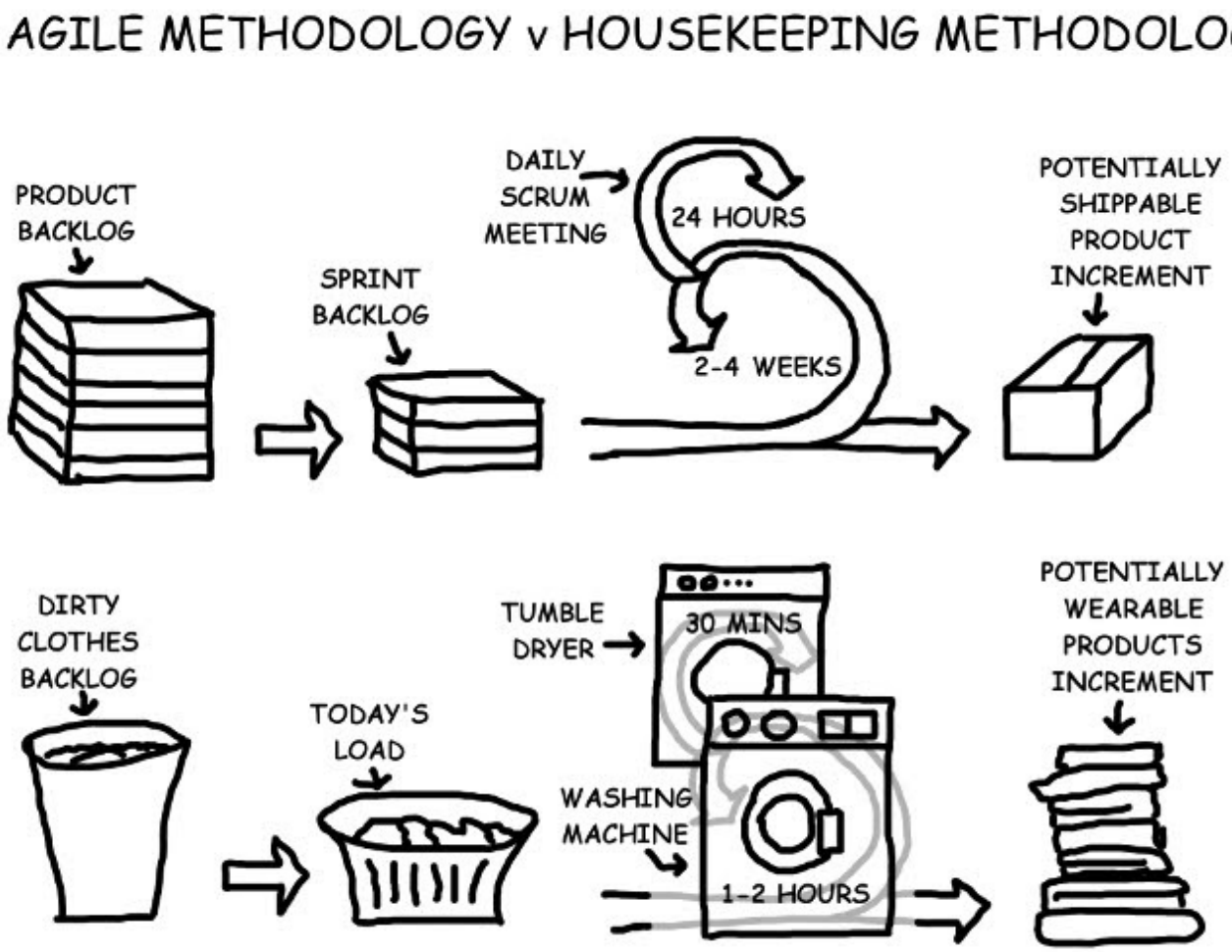
- Agile focuses on producing working software
  - However, Agile *does not* forgo design activities!



- Poor attention to design in Agile yields rapid progress *at first*, but then significant rework is required to scale up and integrate the system.
  - Agile correctly done: Design emerges as sys is developed.

Managing an Agile project: **Scrum**

- **Scrum** (2010): A framework for managing Agile projects

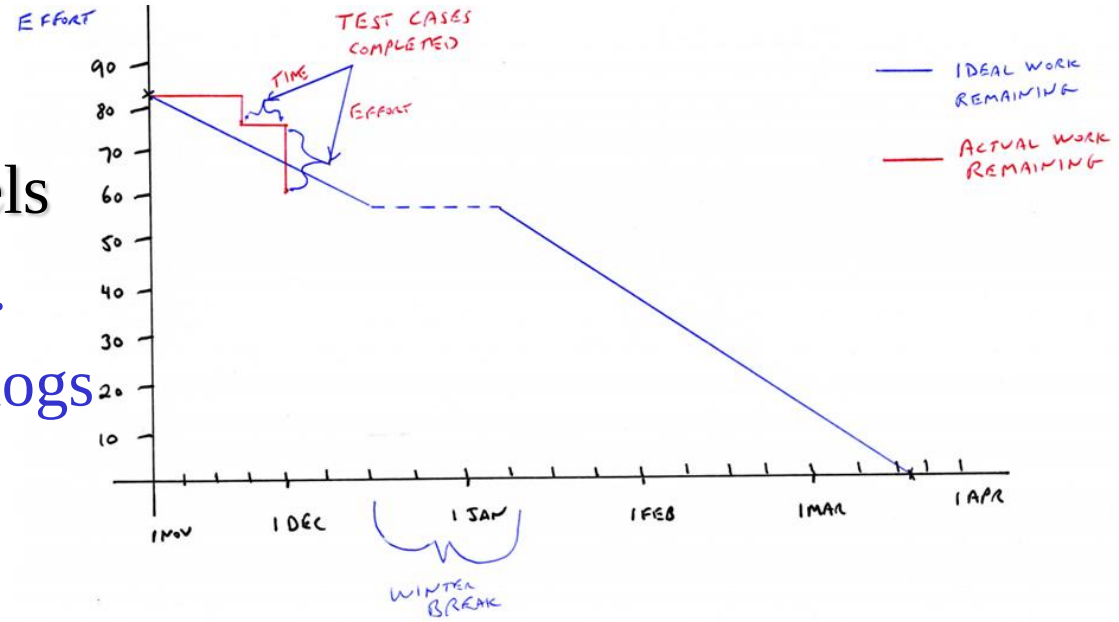


Shippable product gets larger as more and more Sprints are integrated together

Scrum Sprints & Burndown Charts

- **Effort:** Using Fibonacci Sequence (0,1,1,2,3,5,8,13,21,…) assign each acceptance test case a relative "effort" level.
  Why use Fibonacci numbers?

- **Product Backlog:**
  The sum of effort levels
  - Release Backlog vs.
  - *shorter* Sprint backlogs

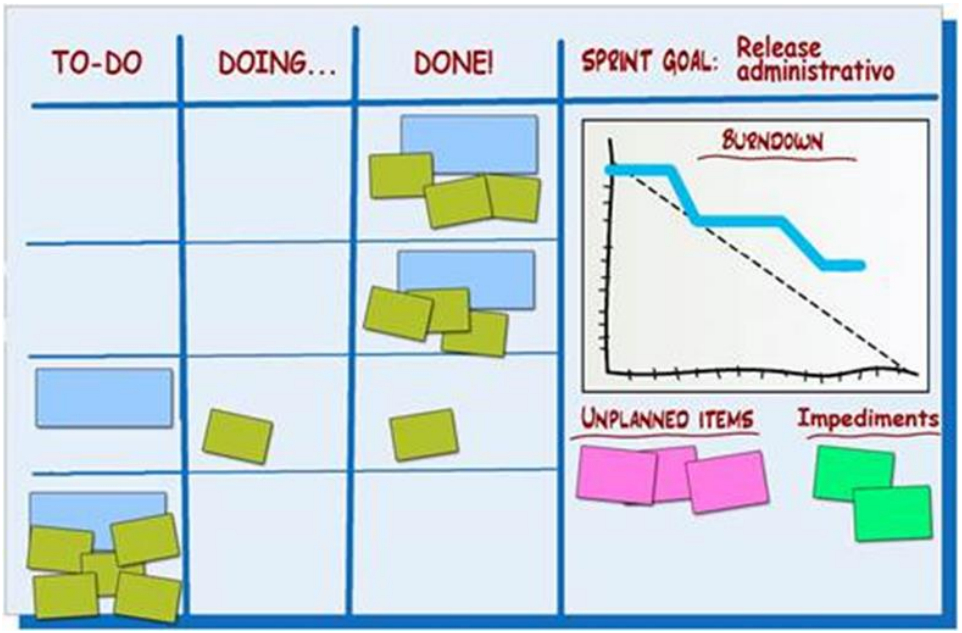


- **Burndown Chart**
  - Ideal Work Remaining Line (starts with Release Backlog Value on left and zero on targeted completion date on right)
  - Plot Actual Work Remaining Line as Customer signs off on test case completion

ICE: Burndown Chart for MIDS 2.0

- Pair up and propose 2 useful things not in MIDS right now.
  - Assign a Fibonacci number estimating the "effort" needed to design, implement, test the requirement. **Uncertainty** also a factor

    (0,1,1,2,3,5,8,13,21,…)



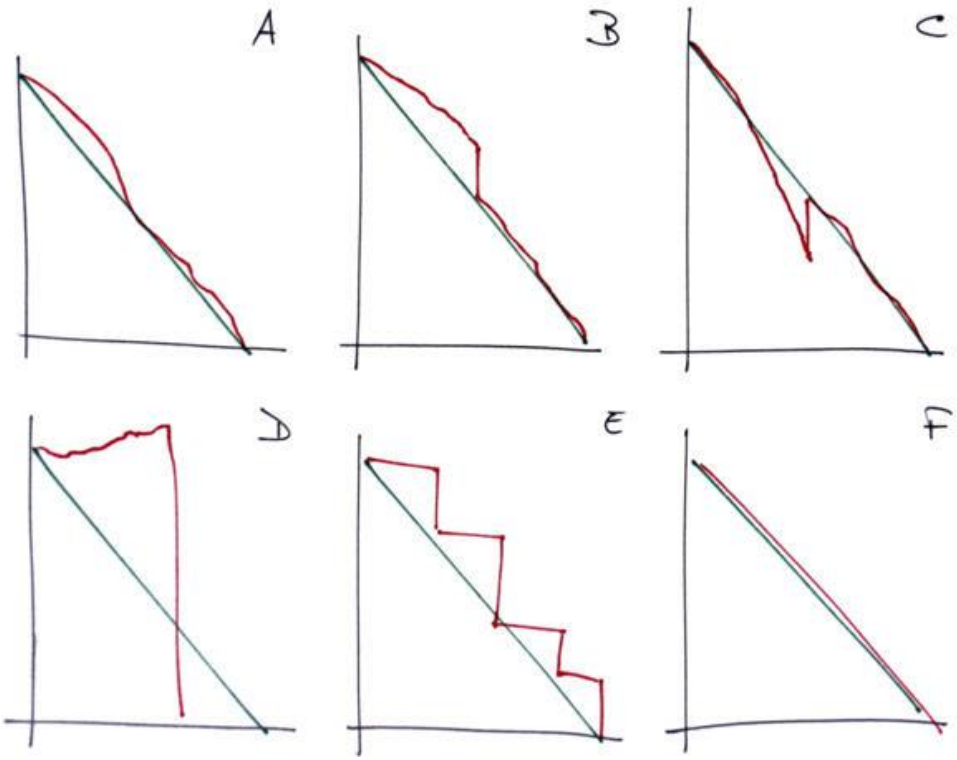
Examples:

- **Multi-factor login** (password + a texted code) might be an effort level of '2'
- **iSearch**: Find a hum/ss course that you have the prereqs for, has a section with open seats, and preserves your current schedule's Wed afternoon Golf class might be a '5'

Note: Consider refactoring high value reqs

ICE: Analyzing Burndown Charts

Match the descriptions to the burndown charts



I. A *(suspiciously)* perfect burndown chart.

II. Tracking well, *frequent* reporting of progress.

III. Tracking well, *infrequent* reporting of progress.

IV. Little early progress, then big push and finished early.

V. Disorganized early in the iteration, big push halfway through to get on track.

VI. Ahead of schedule early on, halfway through Customer wants you to rework something they had previously signed off on.