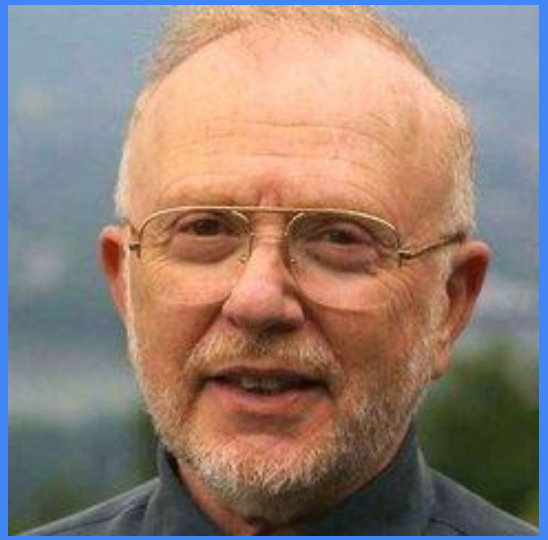
SE101 Prototyping + SShips

Prof Derek Rayside, Associate Director, Software Engineering
Tuesday, October 2nd, 2018

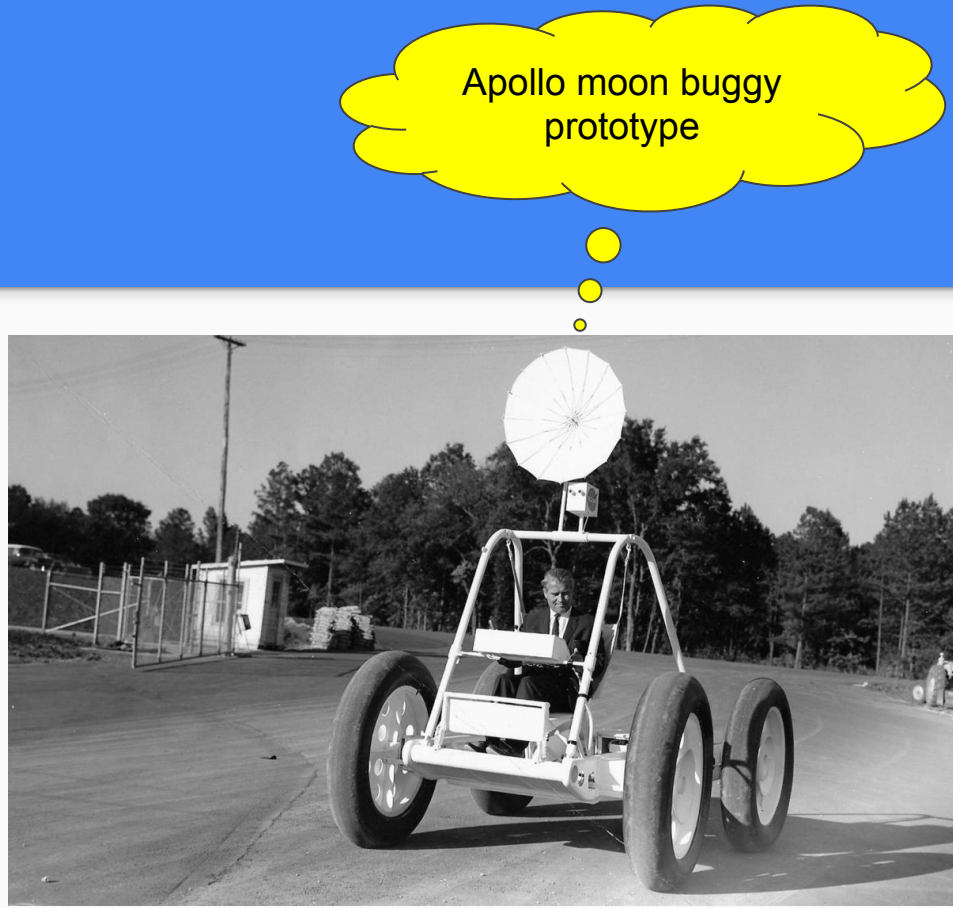
https://docs.google.com/presentation/d/1fn58wOyNUfKb6iJubQ_6g2_gxIvGCPMI5t5D9PPJJkY/edit?usp=sharing

“As a rule, software systems do not work well until they have been used, and have failed repeatedly, in real applications.”
-- David L. Parnas



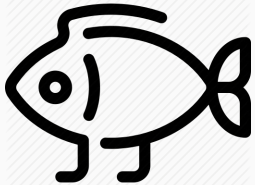
Today's Topics

1. Kinds of Prototypes
2. Spaceships!
    - Learning objectives
    - Commit data
    - Slope vs Y-intercept
    - Git tips
3. First Quest!
4. Co-op Fundamentals



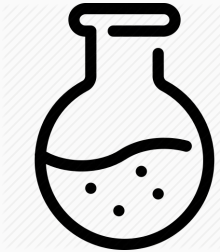
Kinds of Prototypes

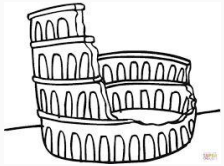
Evolutionary



vs

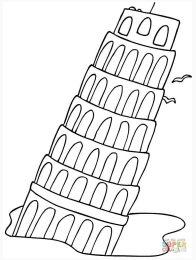
Experimental





Horizontal

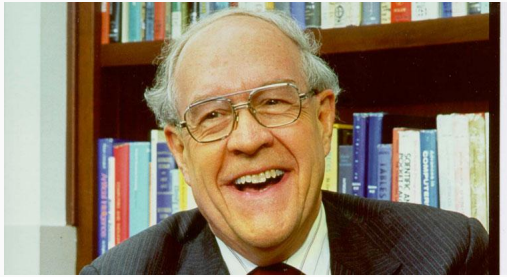
vs .



Vertical

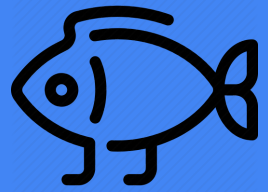
Experimental Prototypes

- Code will be thrown away
- Purpose is to explore an *idea* or *technology*
- Might be written in an alternative language
  - Or using a drawing tool, etc.
- Can *save* dev time!



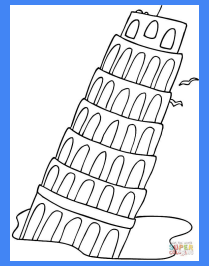
Plan to throw one away.
You will anyhow.
-- Fred Brooks

Evolutionary Prototypes

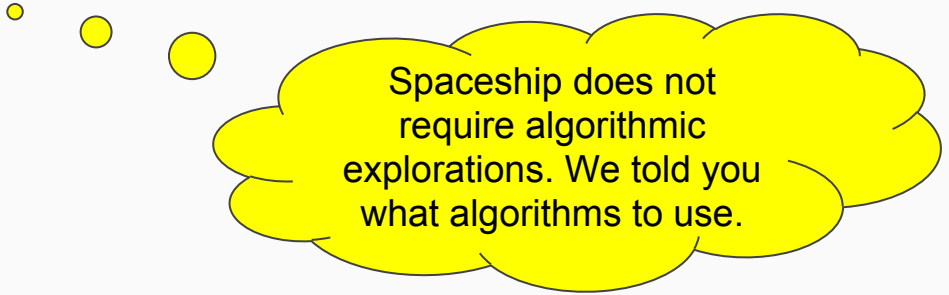


- Prototype evolves to become the real system
- Advocated by ***Agile Methodologies***
- Good when
  - requirements are unclear *or*
  - trying to build a user-base *or*
  - integration is the key challenge
- Not so good for *safety-critical* systems

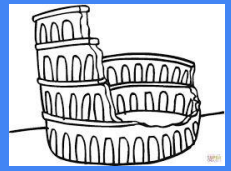
Vertical Prototypes



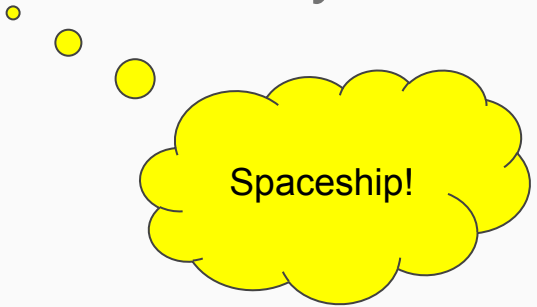
- Explore one issue in depth
- Could be *Evolutionary* or *Exploratory*
- Good for exploring *Algorithmic* ideas
  - e.g., vertical prototypes of algorithms A and B to see performance



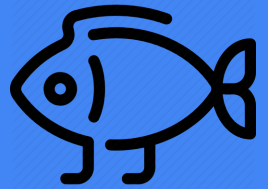
Horizontal Prototypes



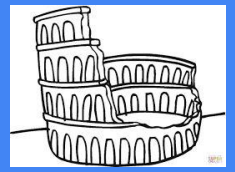
- Shallow exploration of entire system
- Could be Evolutionary or Exploratory
- Good when *integration* is the key challenge



Spaceship Prototyping Strategy



+



Horizontal + ***Evolutionary*** Prototype

- Integration is the key challenge
- Algorithms are known (including stubs/mocks)
- Not safety-critical
- Prototype evolves to become complete system

Spaceship Learning Objectives

Spaceship doesn't count for marks. Too hard to assess the learning: open-ended + team. All your other schoolwork is closed-form + individual ⇒ easy to mark.

Yep

- How to organize software teams
- Pair programming
- Git
- Feature flags
- Horizontal prototyping
- Evolutionary prototyping

Nope

- Object-Oriented Prog. [CS138]
- Dijkstra's algorithm [CS240]
- PD controllers [SE380]
- Unity game engine
- Video game design
- "Hero hacking"

In the real world there will always be some stuff you do not yet know.

Spaceship Trajectories

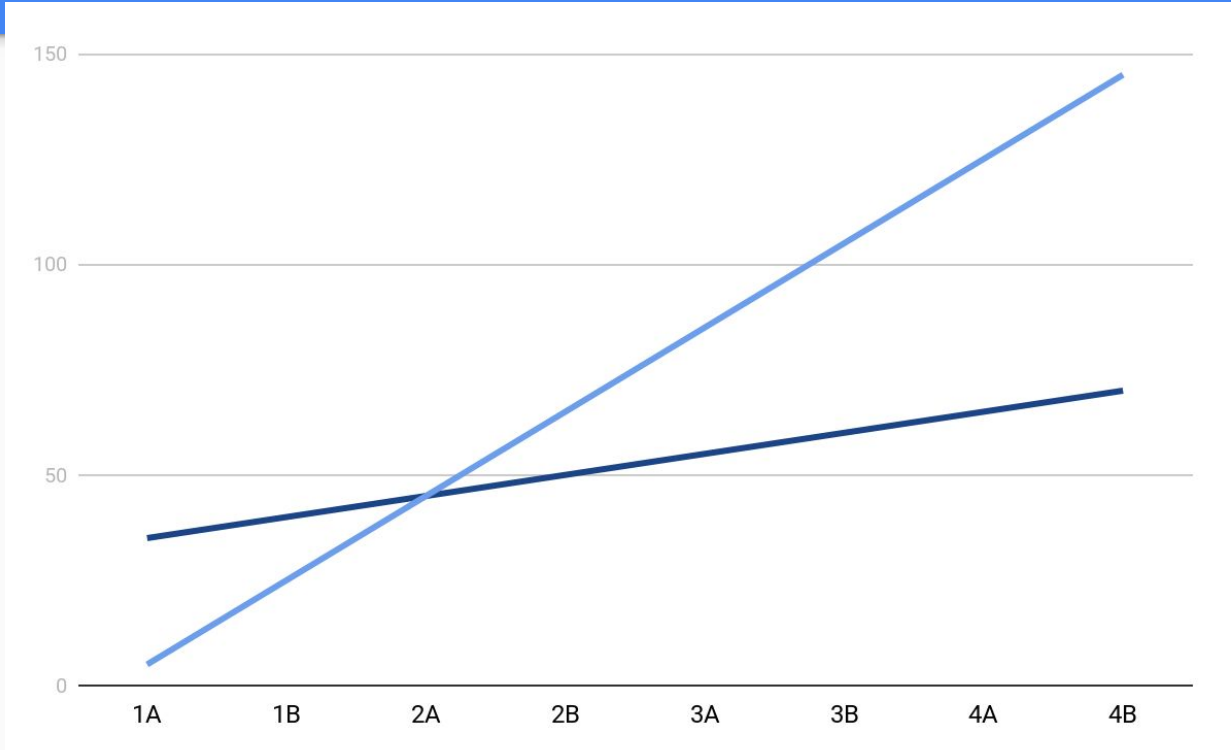
You all have high growth potential. We know it. You need to believe in it, and you need to make the most of your growth opportunities.

Good ***Start***

vs

Good ***Growth***

Always look for growth!



Pity the “Hero Hacker”

- Is there someone on your team who could implement the entire spaceship singlehandedly? ***Probably***.
- Will they learn much from doing that? ***No***.
  - Skills still won’t scale beyond what one person can do.
  - Growth stunted.
  - Stuck in comfort zone.
- Always look for opportunities to grow!
  - Go outside your comfort zone.

Arnold outside his comfort zone

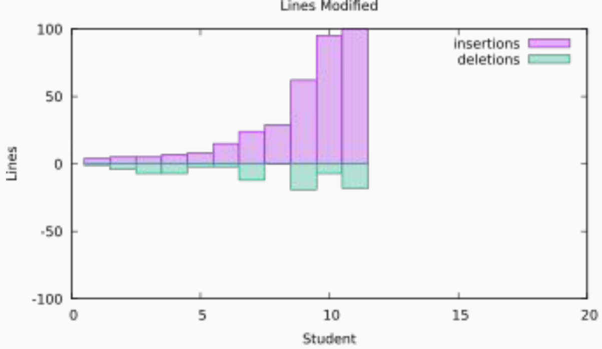
- Was embarrassed by small, weak calves.
- Cut all training pants to show calves.
- Forced himself to confront his weakness.
- Felt uncomfortable. Trained hard. Grew.

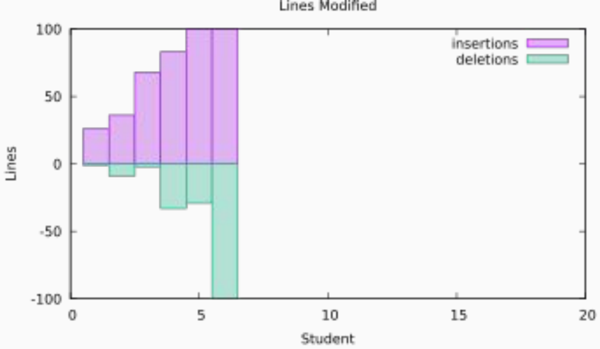
Became Governor of California!

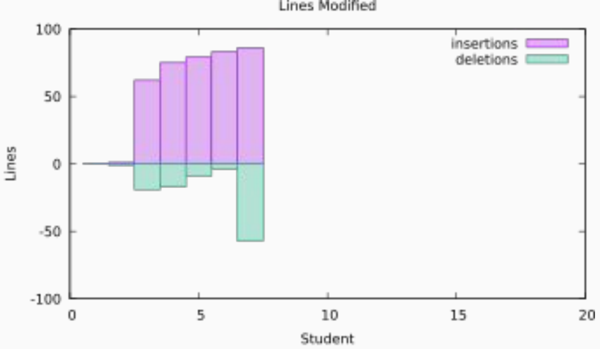
Hayley outside her comfort zone

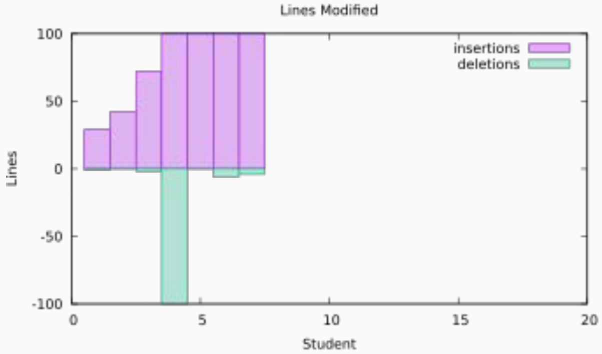
- Most Olympic gold medals of any Canadian: 4 gold + 1 silver
- Greatest female hockey player in world
- Asst. Director of Player Dev for Leafs

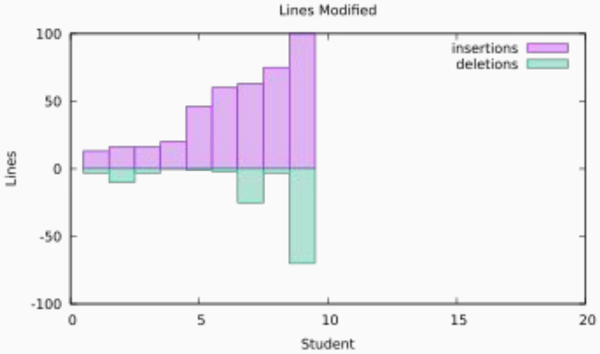
Also attending medical school!

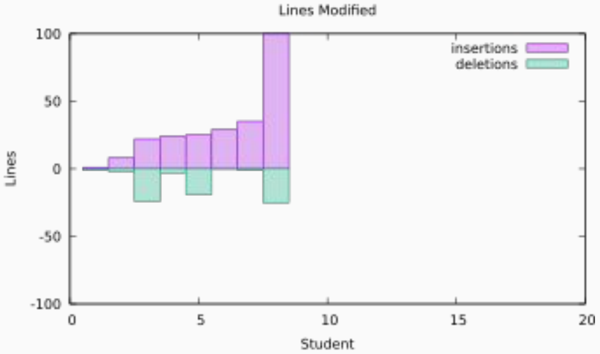


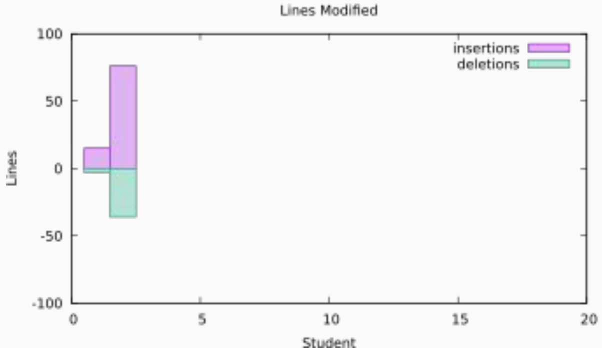












Lines modified by each student

Git Tips

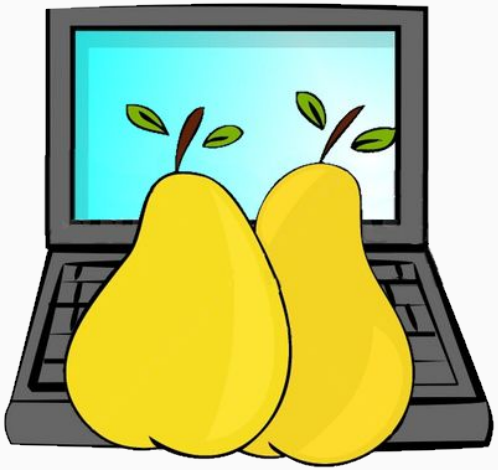
Git is *popular* and *powerful*, but not especially *pleasant* to use. There are other powerful tools that claim to be more pleasant, but are less popular (e.g., Mercurial, Subversion).

- Write meaningful commit comments
- Do not commit all the generated files
  - Use a .gitignore file
- Configure Git on every computer you use
  - Git doesn’t know who you are: you need to tell it on each computer
  - Any settings you made on one computer do not transfer automatically

Pair Programming Tips

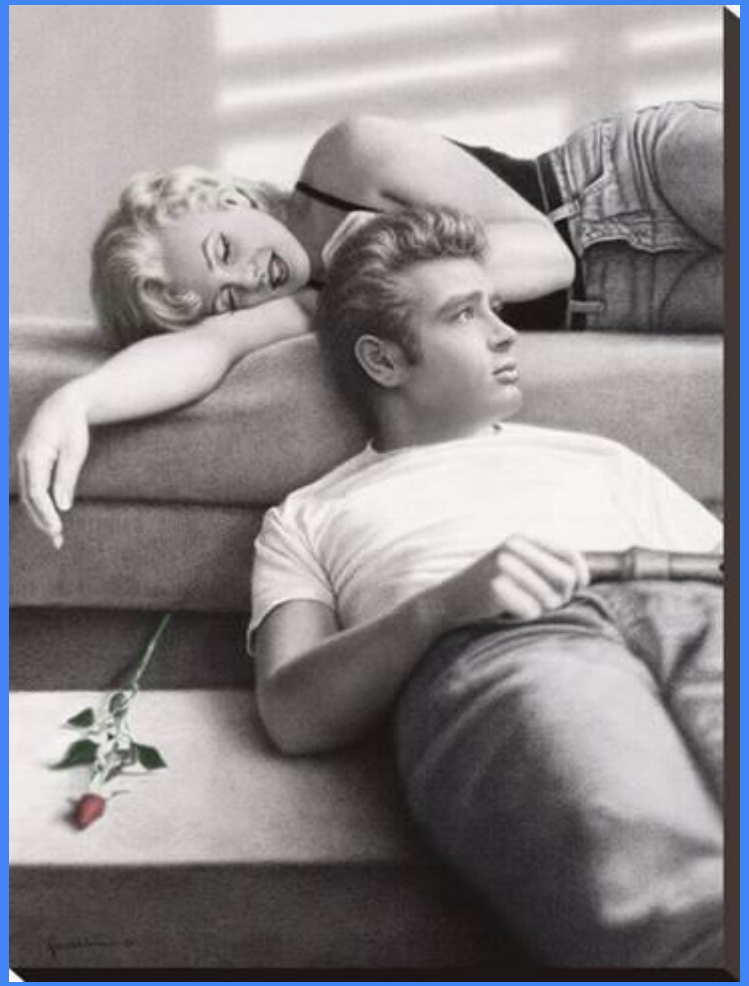
Will you have your first pair programming experiences here at school in an activity that doesn't count for marks, or on the job where your performance matters?

- Discuss ideas on paper first
- Have the person with less programming experience drive
  - Maximizes driver's technical growth
  - Maximizes navigator's soft-skill growth
  - Slope overcomes Y-intercept!



Quest #1: Difficult

Go to bed like your grandparents did.



Quest #1: Difficult

- Screens off at 9pm
- Screens stay off until 7am
- Phones off, in (closed) drawer
- Socialize or study
- Read a book
  - Get a real book!
  - And a real alarm clock
  - And a reading light

- For 5 days
  - Try to be somewhat contiguous
- Log each day
  - Patrick will tell you where (Git)

The scientific point: screen time before bedtime disrupts sleep, which has negative implications for mood, as well as athletic and academic performance. Developing good sleep habits is the foundation of taking care of yourself.