The ACME Climate Project Learning Initiative: A Cheatsheet

Mike Heroux
Senior Scientist
Center for Computing Research
Sandia National Laboratories

Sandia National Laboratories

Exceptional service in the national interest

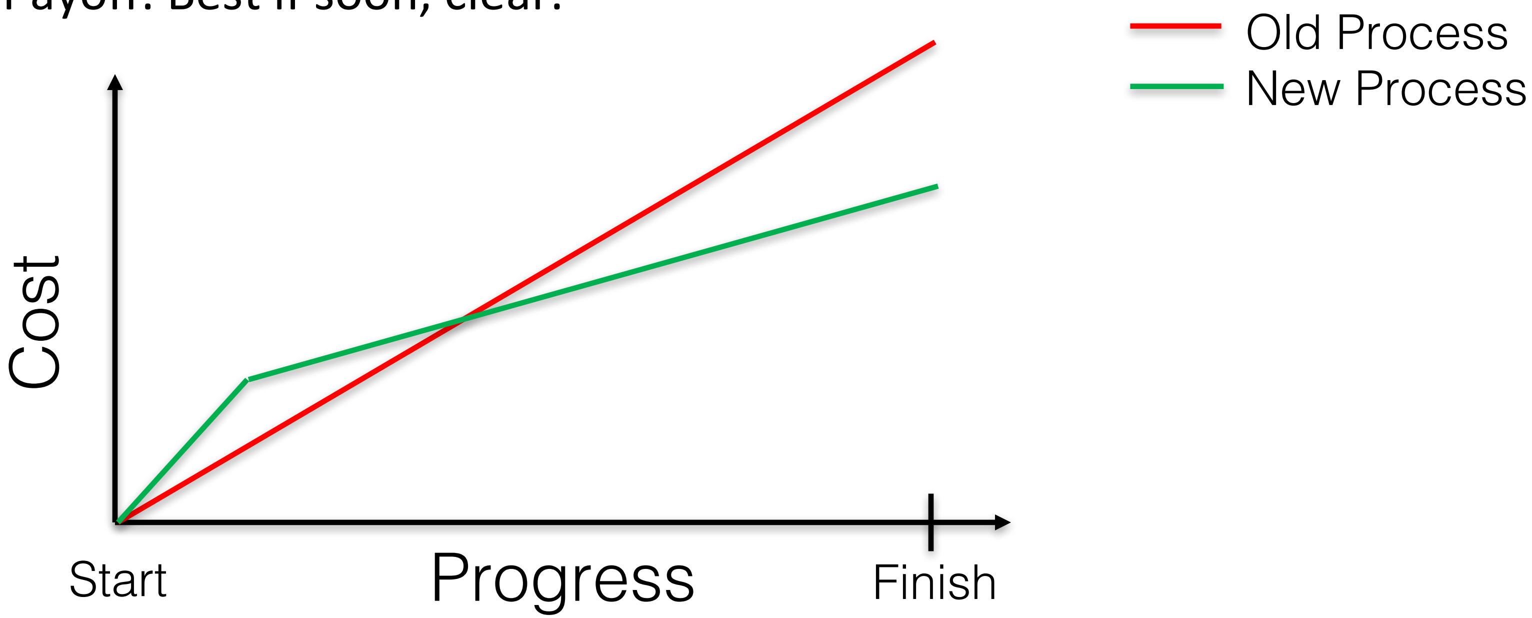
U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

General Strategy

- Interview, analyze, prototype, test, revise, deploy. Repeat.
- Realistic: There is a cost.
  - Startup: Overhead.
  - Payoff: Best if soon, clear.



7 Initial ACME Interviews: Emerging themes

Diverse cross-section: Lab, Component, Proximity to SE Group, Experience.

- **Software challenges compete with other high priority demands.**
  - Urgency of science challenges is paramount.
  - Software improvement must be introduced carefully, with timely and highly probable payoff.
- **Challenge working with Git, especially:**
  - Efficient management of simultaneous development of shared code.
  - Uncertainty with uncommon but essential processes.
  - Uncertain understanding of how Git really works.
- **Testing concerns mentioned often:**
  - Testing process not uniform.
  - No standard test harness.
  - Groups evolve own testing approaches.
- **Unit testing often mentioned:**
  - Desire for quicker, more localized testing, i.e., unit testing.
  - Concern about feasibility of unit testing.
- **Shorten the development cycle:**
  - More features with less work.
  - Fewer merge conflicts.
  - Lower barriers for scientist-developers.
- **Desire for better, more uniform developer training and minimal skill levels.**
  - Basic developer workflows.
  - Coding standards; readable, sustainable source code.
  - Effective commit log messages.
  - Tempered by concerns of too much emphasis.
- **Tools and processes should be kept simple, easy-to-use:**
  - ACME team is diverse, simplicity is important.
  - External collaborators can more easily contribute to ACME and use product.
- **Learning opportunities should be varied:**
  - New team member orientation.
  - Face-to-face, webinars, individual learning plans.
  - On-demand access to software experts.
- **Programming for performance:**
  - Basic performance concepts.
  - Performance portability.
- **Challenges using JIRA effectively, especially in the presence of GitHub issues.**
  - GitHub issues used daily, considered essential.
  - JIRA used less frequently, often an afterthought.

ACME Learning Strategies

Exploring various approaches:

- Real-time, face-to-face? Software Carpentry.
- Real-time, webinar? Coordinate with LCFs.
- Recorded, webinar? By-product of real time.
- MOOC, SPOC? Udacity, etc. Plus local expert.
- Individualized?
  - Slack, On-demand?
  - Github-based?
  - Audible (my favorite way to learn).

Clear learning subject: Git

- Powerful, challenging.
- “Defensive” Git Training
- Teach basic worflows: yes.
- Teach also:
  - Prepare to avoid disaster.
  - Prepare for disaster.
- Practice disaster recovery:
  - Create disaster.
  - Recover.
  - In safe setting.
- How to deliver? To whom?

> *With Git as your source management tool, everyone feels stupid.*
>
> *John Cary*

Summary

- I have experienced the “help” of SW Engineering Experts.
  - Cray circa 1990, ASCI circa 2000
  - Ignored their own process:
    - Failed to elicit, analyze requirements.
    - Slapped on pre-defined solutions.
    - Failed.
- Hopefully realistic: This will not be easy.
- Goals:
  - ID biggest opportunities.
  - Create content and delivery strategies.
  - Work with ACME developers.