Agile Methodologies Redux

Rinku Gupta
Argonne National Laboratory

Software Productivity and Sustainability track, ATPESC 2021

Contributors: Rinku K. Gupta (ANL), Michael A. Heroux (SNL), James M. Willenbring (SNL)

U.S. DEPARTMENT OF **ENERGY** | Office of Science

NNSA National Nuclear Security Administration

License, Citation and Acknowledgements

License and Citation

Acknowledgements

Outline

- Refining our Epic

- PSIP: Productivity and Sustainability Improvement Planning

More on Epic, Story, Task

- Definition of Done

- Refining Issues

- Agile Estimation

Epic, Story, Task Review

- Break down and refine <u>when and as needed</u>
  - Close to when the work will be done
  - Only for work that will take place
  - Can be valuable for estimating
  - There is no "correct" level of granularity

- Epics are very high level objectives

- Stories should represent an increment of value to the customer
  - "Definition of Done" – understandable to user

- Tasks are the steps necessary to complete a story
  - May not individually provide value to the customer

Definition of Done

- Simplified definition: When all acceptance criteria are met

- Acceptance criteria
  - "Conditions that a software product must satisfy to be accepted by a user, customer or stakeholder." – Microsoft Press
  - "Pre-established standards or requirements a product or project must meet." – Google
  - Can include functional, non-functional, and performance requirements.

Definition of Done

- Important to establish for a story before estimating or beginning a task

- Defined by the team, acceptable to customer
  - Customer language

- Should not specify an implementation unnecessarily

Refining Our Epic

- Epic: Refactor code for enhanced modularity
    - Description: The heat equation code needs refactoring to improve modularity. Specifically, there are utilities that could be generalized and used with for other applications. Also, the integration function is currently hard-coded. In the future, we want to use alternative integration functions, so we should generalize the interface for this function.
    - Story 1: Separate out utilities
        - Definition of Done
        - Task list

    - Story 2: Separate out integration function
        - Definition of Done
        - Task list

Refining Our Epic

– Story 1: Separate out utilities

- Definition of Done
  - Unit tests pass
  - Code review completed
  - Integration/system tests pass
  - Utility performance is at least 95% of pre-separation performance
  - Utility usability demonstrated outside of heat equation application

– Story 2: Separate out integration function

- Task 1: Add testing for integration function to protect functionality during refactor
  - Needed testing should be specified
- Task 2: Generalize interface to allow alternative implementations
- Task 3: Expose current integration function through the new interface & run tests

Agile Estimation

- Estimating is hard
  - Requires practice
  - With practice, it is still hard

- Stories are estimated using "story points"
  - Relative estimate
  - Many estimating techniques
  - Should NOT map to hours, days, etc
  - Definition of done needed, tasking not required

- Tasks are estimated in hours
  - Absolute estimate

- Useful for planning schedules

Key concept:
It is easier to accurately estimate many small tasks than to estimate a large epic.

Epic: Huge refactor effort

Tasks:
- Add tests
- Generalize interface
- Expose existing interface

IDEAS productivity

ECP EXASCALE COMPUTING PROJECT

How To Get Better

"Use iteration and incrementation only for projects you want to succeed."

-   Adaptation of Martin Fowler quote

Strategy for Incremental Productivity Improvements

- Identify, analyze, prototype, test, revise, deploy. Repeat.

- Realistic: There is a cost.
  - Startup: Overhead
  - Payoff: Best if soon, clear
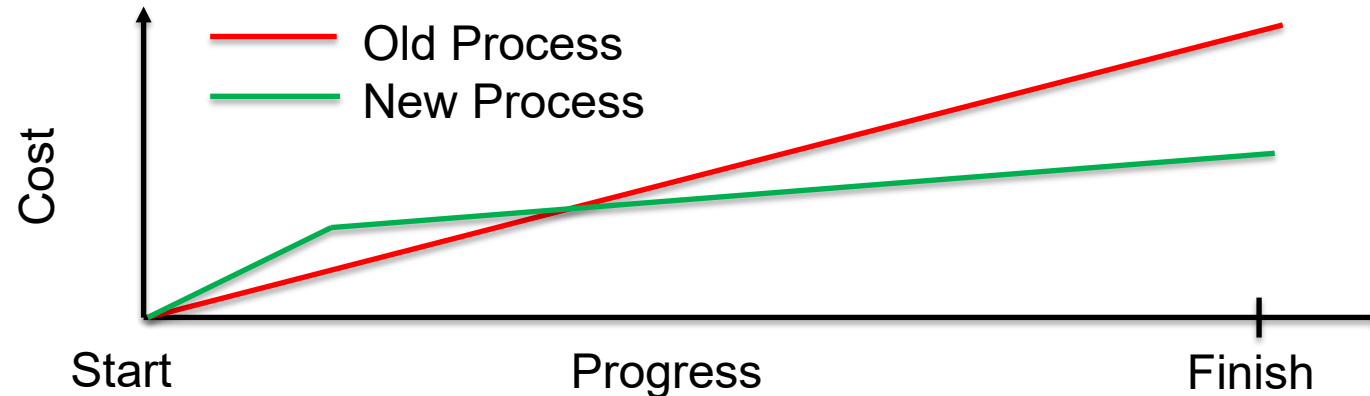
- Working model:
  - Reserve acceptable time/effort for improvement.
  - ***Improve how you do your work on the way to getting it done.***
  - Repeat.

Productivity and Sustainability Improvement Planning (PSIP) Examples: EXAALT & MPICH

PSIP workflow helps a team create user stories, identify areas for improvement, select a specific area and topic for a single improvement cycle, and then develop those improvements with specific metrics for success.

EXAALT PSIP: Continuous integration (CI) testing

BSSw blog article: Adopting Continuous Integration for Long Timescale Materials Simulation, Rick Zamora (Sept 2018)

MPICH PSIP: Onboarding new team members

https://bssw.io/psip