



# ONAP Project Roles and Responsibilities

Chris Donley 2/6/16

Congratulations! Your project was approved

# What now?

# Open Source Behaviors

- Different organizations have different value propositions
  - Generally should be allowed to pursue them
  - Emphasize collaboration
- Build trust with other participants
  - Capabilities and Intentions
  - Be open & welcoming – build relationships
- Fast > good
  - *If you are not embarrassed by the first version of your product, you've launched too late* – Reid Hoffman (LinkedIn)
  - Quality will come in time if we establish a rapid release cadence
- Have an opinion; code is the best way to explain it

# Open source project roles

- Project Technical Lead (PTL) - project leader
  - Could be the person who initiates the project or elected by committers
  - Responsible for planning, project management, recruitment, etc.
  - Also a committer
- Committer – senior technical resource
  - trusted by the project to commit code/requirements changes
  - Usually a senior developer
  - Best practices: 3-5/project
- Developer/Contributor
  - Anyone else working on the project
  - Code submissions must be approved by a committer
- Note that some project roles may be reassigned/delegated, but please notify the TSC and/or release manager if you do so we know who to contact

# The team

- Typically 5-9 people
- Cross-functional:
  - Programmers, testers, user experience designers, etc.
- Members should be full-time
  - May be exceptions (e.g., database administrator)

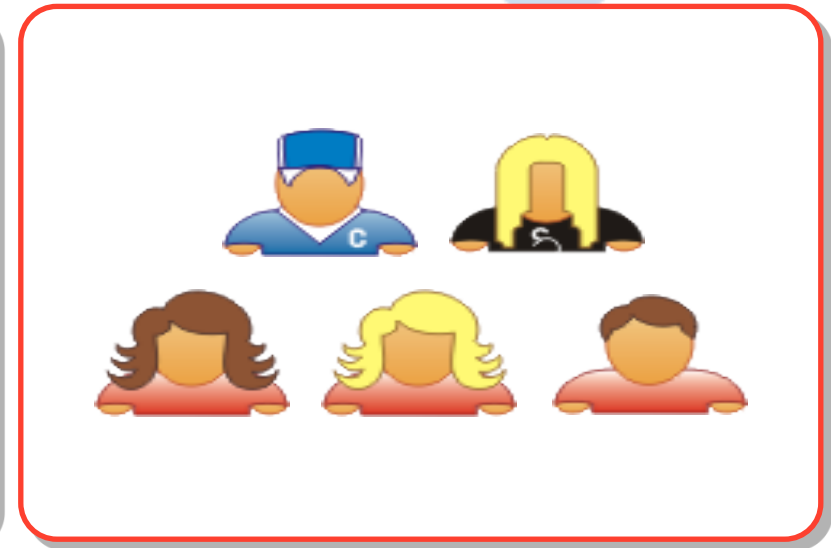
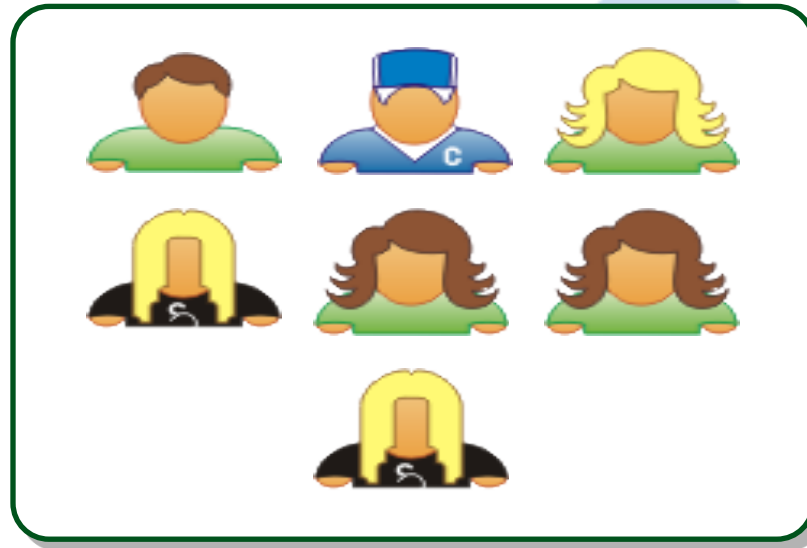
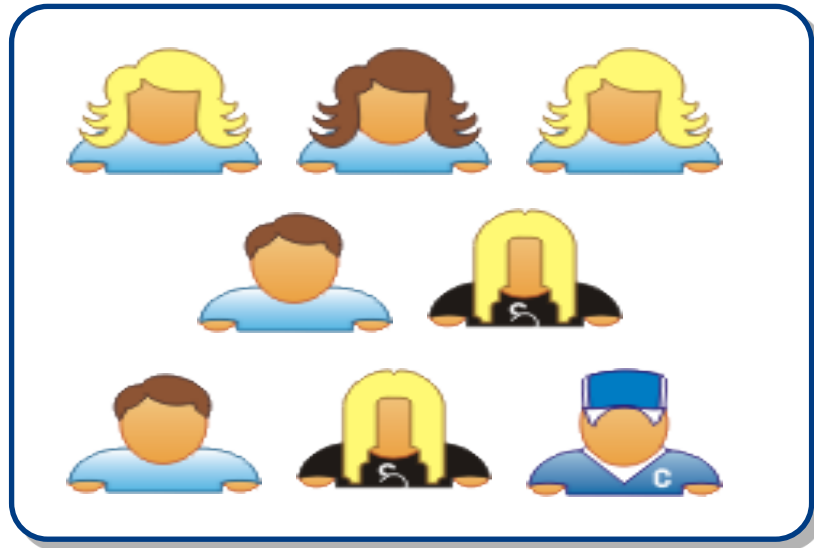


# The team

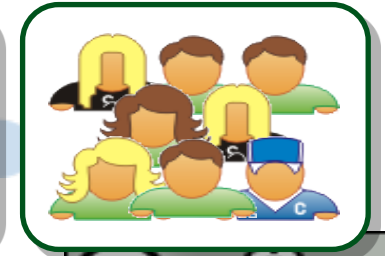
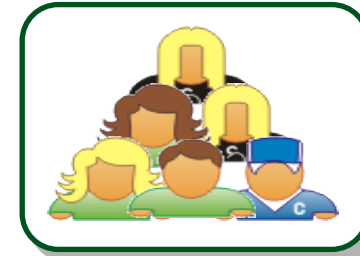
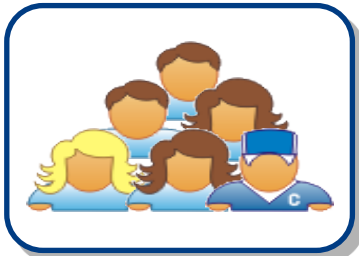
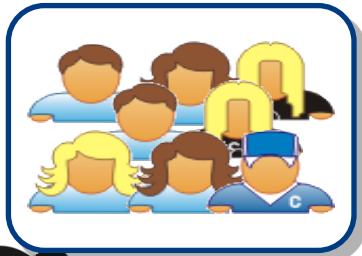
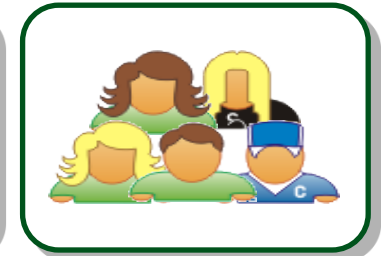
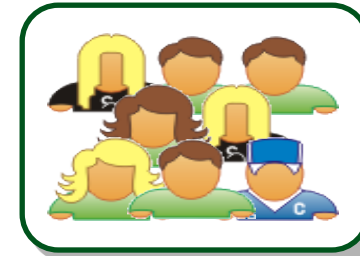
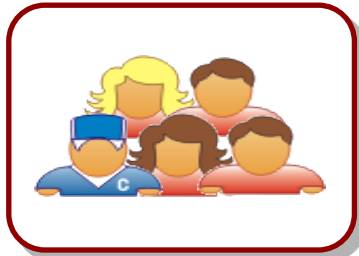
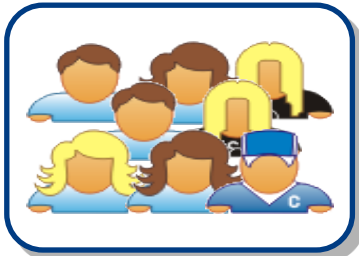
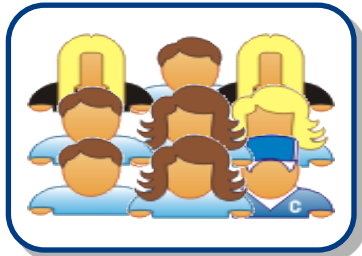
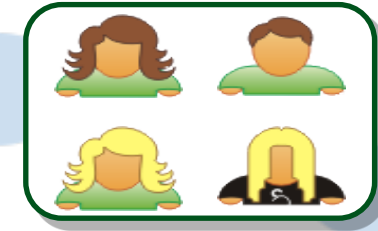
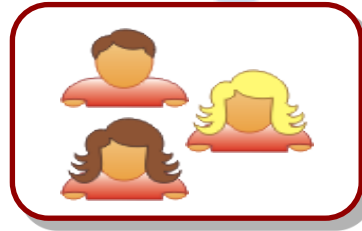
- Teams are self-organizing
  - Ideally, no titles but rarely a possibility
- Membership should change only between sprints



# TSC <-> Projects: Scrum of scrums



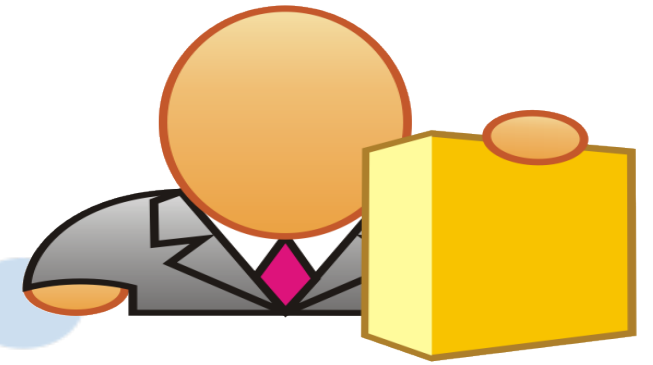
# Sub-projects: Scrum of scrum of scrums





- Overall lead for the project
- Responsibilities:
  - “Project Owner” – maintain, prioritize and communicate business requirements (with committers)
  - “Scrum master” – manage development cycle, maintain Jira status, manage burn-down list, etc.
  - Recruiter – recruit developers to work on your project
  - Point of Contact – communicate with TSC and interested stakeholders

# Product owner



- Define the features of the product
- Decide on release date and content
- ~~● Be responsible for the profitability of the product (ROI)~~
- Prioritize features according to market value
- Adjust features and priority every iteration, as needed
- Accept or reject work results

# The ScrumMaster



- Represents TSC to the project and vice versa
- Responsible for enacting Scrum values and practices
- Removes impediments
- Ensure that the team is fully functional and productive
- Enable close cooperation across all roles and functions
- Shield the team from external interferences

# Expectations for PTLs

- Represent the team to the TSC (and vice versa)
- Communicate project vision internally and externally
- Guide the team in creating and prioritizing backlog
  - OK to divide into sub-teams (each led by a committer)
- Manage and report on project status
  - OK to delegate, but let us know so we know who to speak with
  - Identify any blocking areas where you need help
- Host regular team meetings
- Prioritize on-time delivery
- Encourage collaboration and manage conflicts
- Recruit more developers and ensure existing developers feel valued and don't drop out.

# Committer (typically 3-5/project)

- Serve as part of the “brain trust” for the project
- Contribute to strategy and prioritization
- Vote on project-level issues
  - PTL election
  - Committer elevation
  - Project-specific technical, planning, or logistical issues
- Develop/commit code and tests
- Code reviews

# Developer/Contributor

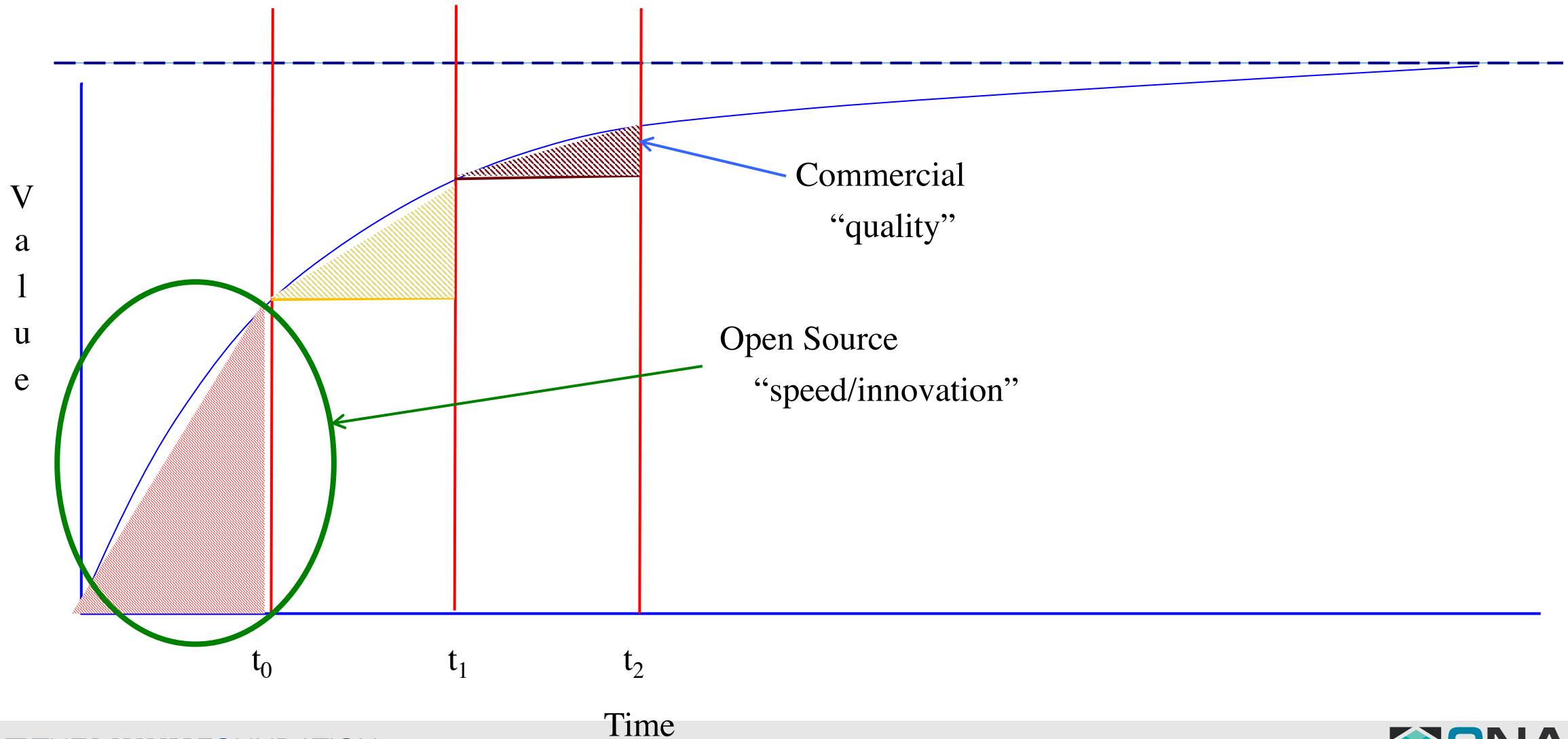
- General project participant
- Develop code, models, and/or documentation
- Help with testing
- Participate in other project activities

# Development process

- We generally follow Lean/Agile development
  - Rapid development cycles
  - Constrained scope
  - Testing/Feedback
- Why Lean?
  - Help focus on the right problem
- Why Agile?
  - People are generally bad at forecasting/planning
  - Breaking work down into smaller pieces improves accuracy
  - Quickly react to changing technical/market environments

# Law of Diminishing Returns

Expectations on Quality vs. Speed





# Scoping our Releases

- Speed vs. Efficiency



Fast



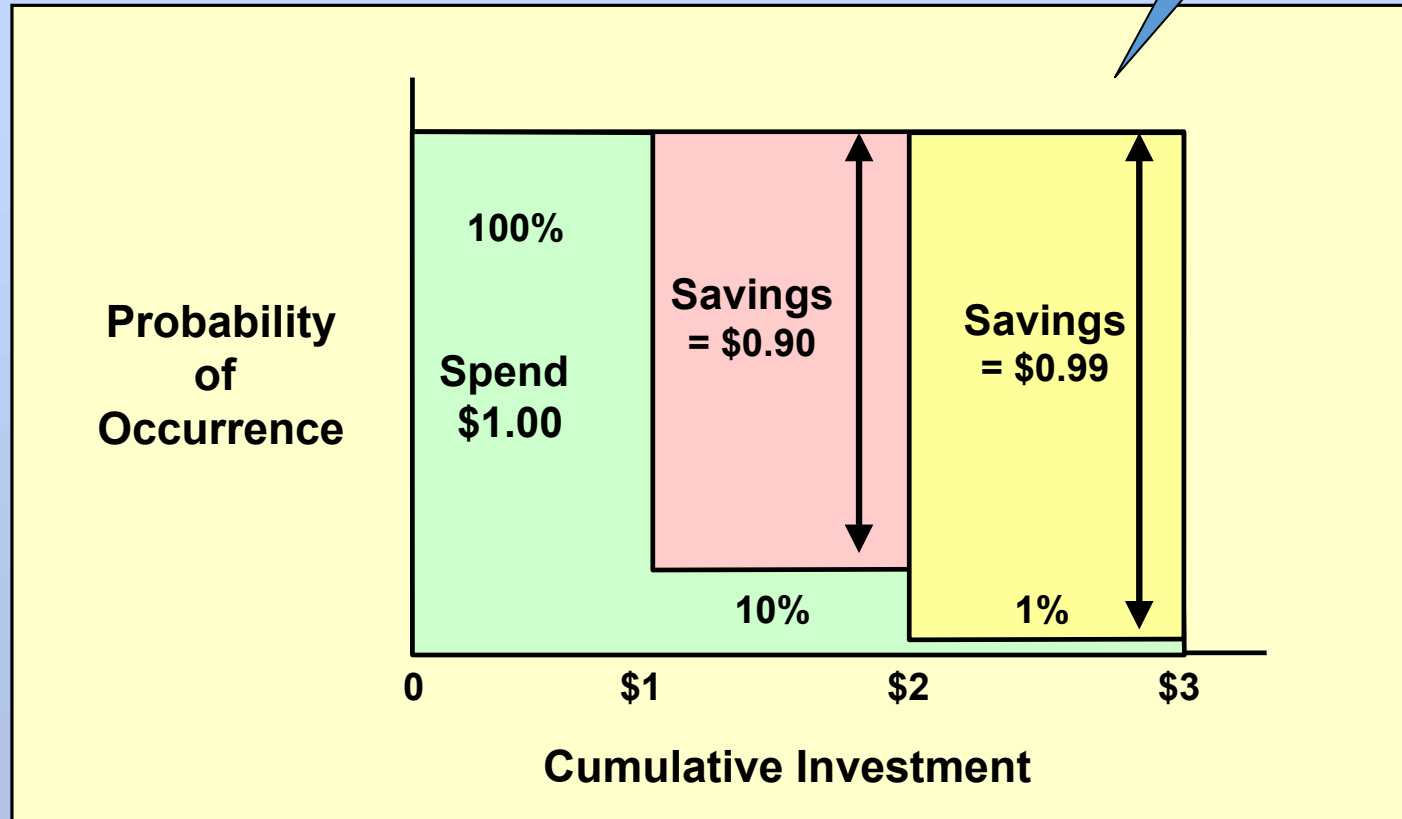
Efficient

# Shortening our Cycle time

- Rapid Feedback - Lottery Example

\$1.11 vs. \$3

## Value of Feedback



In this case, accelerated feedback reduces required investment by 63 percent.

# Why all the milestones? Cadence/Synchronization

- Small changes in arrival time can multiply delays if synchronization is poor
- Cadence can help minimize waiting
  - More frequent opportunities reduce impact of a 'miss'



# Next steps

- Establish regular team meetings
  - Weekly?
  - Daily standup?
- PTL elections
- LF will set up repos and jira space
- Put together your release plan
  - Identify minimum viable product (MVP) and stretch goals
  - If you get in trouble, what can you cut?
- Create jira tickets and plan your first sprint
- Fill out the release planning template and other information on the wiki
- Execute!



**ONAP**

OPEN NETWORK AUTOMATION PLATFORM

谢谢