



# The Future of Testing: How Community Engagement Is Changing the Rules

Aditya Dada (Sun Microsystems)

Varun Rupela (Sun Microsystems)

# Agenda

- Open Source is here to Stay. Why ?
- The Problem
  - > Quality Challenge
- The Solution
  - > Open Sourcing Quality
    - > Logistics for Participating in an open source project
    - > Logistics for Managing an open source project
  - > A Win-Win solution
  - > A Shift in Paradigm
- The GlassFish Quality Community Project
- References

# Open Source is here to Stay. Why ?

- Choice – Good and Bad
- Differentiators
  - > Whats New, Cool and Useful - *Innovation*
  - > Does it Work ? *Quality*
  - > Is it Easy to Use ? *Quality*
- *QUALITY*
  - > Day 1 – What Goes in ?
  - > Continuous Process
- Deliver a Great Product !!

# Open Source is here to Stay. Why ?

- Community Driven
  - > Need Based
  - > Priority
  - > Better Relationships with Customers
- Innovation happens else where (PHP, Ruby on Rails)
- Adoption and Visibility
- Access to Newer Markets
- Build Brand Loyalty for Company Hardware
- Improved Quality !!

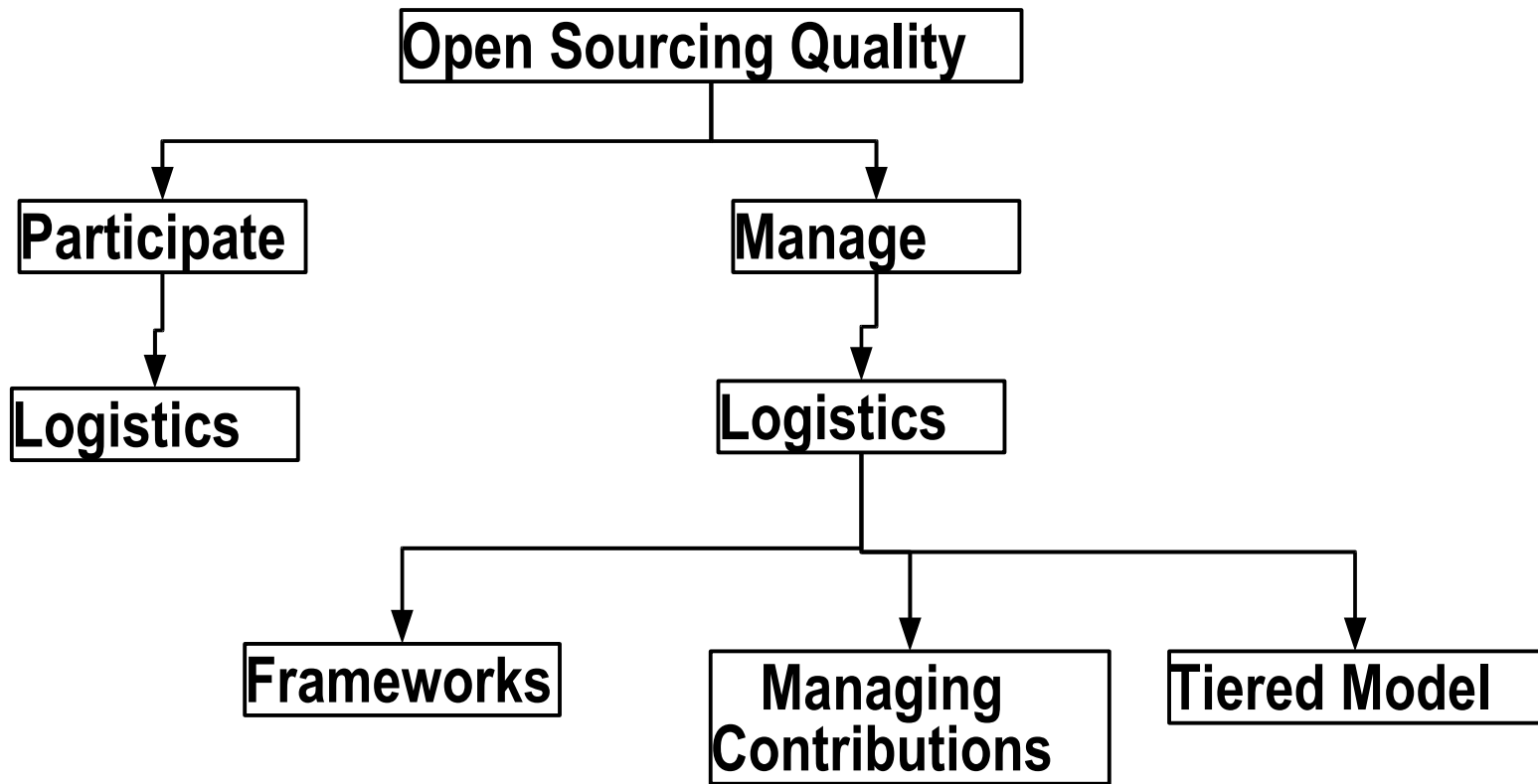
# The Problem

# Quality Engineering Challenge

- Traditionally
  - > In-House Test Development
  - > In-House Test Runs
- Problem of Plenty
  - > Many New Features Added Quickly
  - > Example - Glassfish V3 supports – Scripting, Ajax ...
  - > Used in Different Ways – Wiki Server, Plugins ...
- Same QE Resources
- *Greater Quality, Beat Competition ?*

# The Solution?

# The Solution



# Open Sourcing Quality

- Contributing to an open source project
  - > 1000s of projects to choose from
- Converting your project to open source
  - > Take your source out to community
  - > Create processes
  - > Move existing developers external
- Starting your own project
  - > Create basic usable project
  - > Evangelize to attract developers
  - > Create processes to manage team

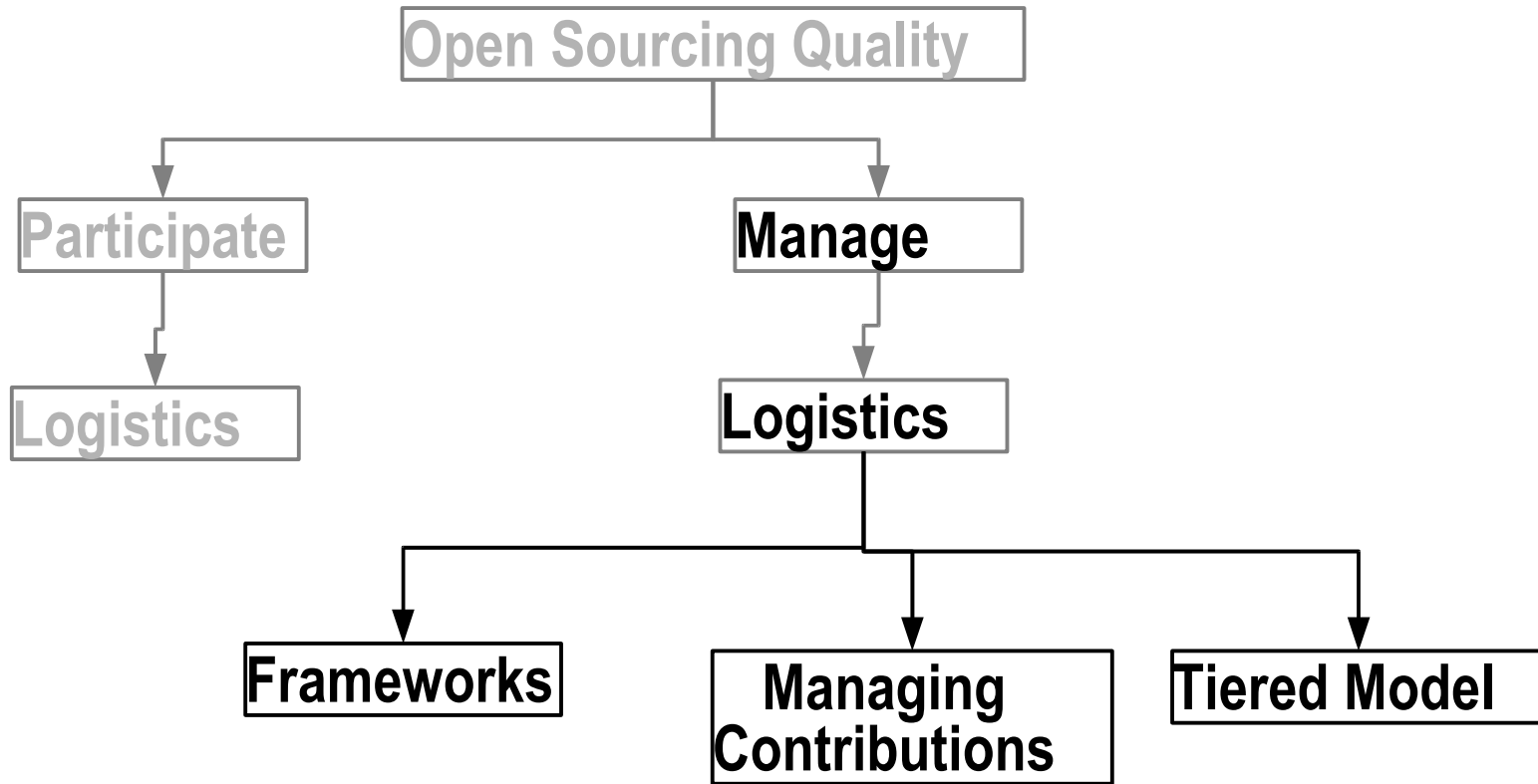
# Logistics of Participating in OSS

- Get required permissions
- Learn about product features, code & bugs
- Create patch to fix a bug & submit to the committee
- Develop a trust relationship
- Gain commit rights
- Enhance, develop and fix features of the product

# Logistics for Managing an OSS project

- OSS is based on “Common Functionality”
  - > Everyone wants it
  - > Vendors “Differentiate”
- Engaging the community
  - > Rewards: Learning, Contests, Networking
  - > Improving Communication
    - > Blogs, Email aliases, Articles, Talks
- Helping community contribute
  - > Creating frameworks
  - > Managing community contributions
  - > Harnessing a Tiered Model

# The Solution



# Creating Frameworks

- Why do you need a Framework?
- Framework Design Decisions
  - > Platform Independence
    - > Java / XML
  - > Database Independence
    - > Separate DDLs for each DB
    - > DAO Design Decision
  - > Extensible
  - > Common Configuration & Services

# Sample Framework Design

- Workspace Design
  - > wsroot
  - > wsroot/config <=Common Files
  - > wsroot/build <=Compilation/Packaging
  - > wsroot/util <=Libraries/Helpers
  - > wsroot/<components>
  - > wsroot/ejb/cmp
  - > wsroot/web/servlet
- Different <wsroot> for different teams, components.
- Common targets, properties controlled from top.

# Sample Framework Implementation

- ANT Based

- ```
<target name="deploy-common" depends="init-common">
  <property name="as.props" value="--user ${admin.user}
  --password ${admin.password} --host ${admin.host} --port
  ${admin.port}"/>
    <exec executable="${ASADMIN}" failonerror="false">
      <arg line="deploy"/>
      <arg line="${as.props}"/>
      <arg line="--name ${appname}App"/>
      <arg line="--retrieve ${assemble.dir}"/>
      <arg line="${assemble.dir}/${appname}App.ear"/>
    </exec>
  </target>
```

- > This is a snippet of a simple target to deploy an EAR file to the application server. In our e.g., this is common functionality used by most components.
- > This is kept in a 'common.xml' file under the 'config' directory that contains all the common and shared code.

# Managing Community Contributions

- Creating a Transparent Process
  - > Transparency is important!
  - > Communication is most Important!
- How to Encourage Contributions
  - > Market ideas differently to different types of contributors
    - Put yourself in their shoes
  - > Make it Rewarding
- Tips on Critiquing Entries
  - > Open Code Reviews by Community & Employees
  - > Organize submissions by features
  - > To Accept or Not To Accept – Broken applications?

# Harnessing a Tiered Model

- Testing is Expensive!
  - > Test to see if build is ready to be tested.
  - > Identify DOA builds/components
  - > Minimal Depth, Maximum Breadth
- Release Models (Catching Defects With...)
  - > Tinderbox (...every Integration)
  - > Nightly (...every Night)
  - > Weekly (...every Week)
  - > Milestone (...every Milestone)
- Fast Execution, Flexible Reporting, In-Built Services
- Model applies to ALL Products & Quality projects!

# A Win-Win Solution

- Rewards for participating in Open Source Projects
  - > Professional Rewards
    - > Becoming a “star”
      - Visibility is a Key. People working for free. Now that's Visible!
      - Gaining new skills to understand the complete picture
        - Coding, managing, marketing etc.
    - > Become more employable
      - Sun has hired many active contributors.
  - > Personal Rewards
    - > Choice of work you like.
    - > Learning New Skills
    - > Learning from Industry Experts – for Free!

# A Win-Win Solution (Contd.)

- Completing the Eco-System
  - > The companies win by:
    - > Getting free, high quality work
    - > Getting loyal contributors
    - > Get to see talent before employment
  - > Contributors win by:
    - > Enhancing their professional and personal lives
    - > Earning rewards through contests
    - > Networking with experts to
      - succeed in current work
      - Attract developers for new projects
- Its a Win-Win Solution

# A Shift in Paradigm

- Traditionally:
  - > Companies employ engineers to test code
  - > Almost all code is created in-house
- Going Forward:
  - > Companies employ engineers who attract contributors
  - > Most of the test code is created externally
- Paradigm Shift:
  - > Quality Test Engineer skills change from
    - Learn automation techniques, code and test applications
  - > To
    - Evangelism, Sales, Marketing, Direct test & automation efforts, engage community etc.

# The GlassFish Quality Community (GFQC) Project

- What we do and Why
- Infrastructure is in place
  - > Frameworks for contributing easily
  - > Industry experts help with submission critiques, networking, and education.
- 50 new contributors in 2 days!!
- Looking for more contributors!



# GlassFish QUALITY COMMUNITY

GlassFish Quality Community URL: <http://glassfish.dev.java.net/quality>



## > LEARN

- > JavaEE Technologies
- > Latest technologies in GlassFish like Web2.0, EJB, AJAX, JRuby, PHP, Scripting, Social Software, by using and playing with GlassFish test applications
- > Advanced features like Clustering/High Availability of GlassFish
- > Best Practices for Coding End-to-End Applications (Blue Prints/Samples)
- > Improve effectiveness of your developer unit tests
- > Software Engineering processes
- > Mentoring for University projects



## > PARTICIPATE

- > Start participating today by
  1. Joining the Sun Developer Connection at [developers.sun.com](http://developers.sun.com)
  2. Subscribing to [quality@glassfish.java.net](mailto:quality@glassfish.java.net)
  3. Signing Sun Contributor Agreement
  4. Start contributing
- > Help improve GlassFish by
  1. Developing new end to end applications and adding to GlassFish code base
  2. Subscribing to [quality@glassfish.java.net](mailto:quality@glassfish.java.net)
  3. Finding & submitting defects/bugs/issues and enhancements
  4. Sharing your experiences through blogs, articles & tips
  5. Evangelizing GlassFish



## > REWARDS

- > Connect with technology/domain experts
- > Enhance your enterprise Java and software engineering skills
- > Become a contributor for a major open source project like GlassFish easily
- > Earn a spot on the GlassFish Quality Community page
- > Become eligible for general GlassFish GAP awards, 1st prize 20K, 2nd prize 10K, 3rd prize 10K, Base prize 4K
- > Become eligible to participate in quality specific iTouch gift contests

# References

- GFQC: <http://glassfish.dev.java.net/quality>
- <http://perens.com/works/articles/Economic.html>
- Join Java.net: <https://www.dev.java.net/servlets/Join>

# Q & A

Aditya Dada ([aditya.dada@sun.com](mailto:aditya.dada@sun.com))

Varun Rupela ([varun.rupela@sun.com](mailto:varun.rupela@sun.com))