Leveraging community and Adopt a JSR

Martijn Verburg (@karianna)

On behalf of the LJC and the global JUG program

Slide design by http://www.kerrykenneally.com

Acronym soup

- JUG - Java User Group
- JSR - Java Specification Request
- JCP - Java Community Process
- EG - Expert Group
- EC - Executive Committee
- RI - Reference Implementation
- TCK - Technology Compatibility Kit
- OSS - Open Source Soft

JSR-348 has changed the landscape

- JSRs under JCP 2.8 process must be run:
  - Openly
  - Transparently

- This is a good thing. Projects run in a OSS manner:
  - Can have resources that rival large corporations
  - Can be incredible fun to work in (no TPS reports!)
  - Can be more balanced

- But how to achieve those goals?
  - Apply techniques from successful OSS projects
  - Guide volunteers in the right direction

Getting volunteers

- People will not just show up, you need to reach out!

- Individual members and organisations of the JCP
  - Including EC members
  - Send an email to pmo@jcp.org

- Individual members and organisations of the EG

- JUGs - Adopt a JSR program

- Social media and conferences
  - Twitter + conference talks can create real buzz
  - Google+ and Facebook pages help

- Remember to follow up!

Keeping volunteers

- This was covered in a previous presentation, but some highlights:
- Have a policy of “Be Nice”
- Make it brain dead easy to build RI’s and TCKs
- Have a clear front page with links on how volunteers can help
- Have a wiki so users can update info directly

What volunteers can work on

- Remember that volunteers come in all shapes and sizes
- Best to have technical and non-technical tasks
- Have tasks split into levels of expertise
  - Starter
  - Intermediate
  - Advanced
- By having volunteers work through the levels:
  - You’ll get better signal to noise ratios
  - Community will self police and self train
  - Volunteers will feel wanted!

Starter level - I

- Volunteers at the starter level can:

- Test the early Reference Implementation builds
  - Use them to find pain points (“It’s just too laborious to construct X”)
  - Report bugs (“Arggh NPE!!”)
  - Suggest feature enhancements (“A `convertX` method would help”)

- Help triage issues
  - Reproduce issues
  - Erase/merge duplicates
  - Set priorities/categories etc

Starter level - II

- Volunteers at the starter level can:

- Give feedback on design
  - Remind them that semantics is more important than syntax!
  - They can discuss issues with their organisation and deliver feedback
  - They can think about how they would use the JSR in day to day dev

- Help moderate the mailing list
  - Help the community self police towards helpful conversations
  - Pour water on flame wars etc

Starter level - III

- Volunteers at the starter level can:

- Help evangelise the JSR
  - Social media (Twitter, Facebook et al)
  - Blogging (write a post about the JSR)
  - Lightning talks (give talks at user groups or online!)

- Improve project infrastructure and JSR visibility
  - Help setup canned hosting (java.net, GitHub etc)
  - Help with SEO of website
  - Make sure downloads, mailing lists and issue trackers are easy to find
  - Help maintain the FAQ/Wiki

Intermediate level - I

- Volunteers at the intermediate level can:

- Help the JSR meet the transparency principles of JSR-348
  - Check for the public issue tracker
  - Check for EG public communications
  - Check for balanced EG
  - Check that std is a coming together of competing implementations

- This is especially important to ensure the JSR gets positive votes

Intermediate level - II

- Volunteers at the intermediate level can:

- Help build the RI
  - Get coding with the actual implementation of the spec!
  - Best to slice off low hanging fruit to start with
  - Can begin with static code analysis, javadoc etc

- Help build the TCK
  - All implementations must pass this crucial test suite
  - Volunteers to gain real TDD/Unit/Integration test experience

Advanced level

- At the advanced stage, volunteers can:
- Join the Expert Group (EG)
  - They need to be an expert in this technology
  - EG members are central to pushing the JSR forwards
  - High time commitment
  - Lots of personal, community and career benefits

An important note

- Volunteer enthusiasm and skill sets are what JSRs need!

- But their efforts should have focus
  - Sometimes well meaning enthusiasm can be counter productive
  - e.g. They're not all language designers!
  - However, there is always a truly productive way to help a JSR

- Don't forget, co-ordinate and communicate your efforts
  - With the EG and the volunteers
  - With other JSRs

- http://java.net/projects/jugs/pages/AdoptAJSR#Laser-like_focus

Have doubts?

- You only have to look at the incredible success of
  - Ubuntu
  - Jboss community projects
  - Open Office
  - Jenkins

- Leveraging volunteers effectively is non trivial
  - It’s a social problem
  - Get EG members and other volunteers to help with the admin

- See Karl Fogel’s seminal piece at http://www.producingoss.org

What is the Adopt a JSR program?

- See http://adoptajsr.org for full details
- A JUG lead initiative to improve standards in the Java ecosystem
- Groups of JUG members work on JSRs

Why are the JUGs getting involved? - I

- To move the entire Java ecosystem forward
  - To become part of the solution as opposed to staying silent
  - Help define the future
- To be at the forefront of new technology
- To ensure that ivory tower standards do not occur
- To make sure the useful libraries and APIs get built
- To boost the careers of JUG members
  - Gain new technical and community skills and much more!

Why are the JUGs getting involved? - II

- To demystify the JCP and the standards process
  - Get day to day developers knowledgable about the process
- To acquire new knowledge
  - Gain valuable technical skills
  - Learn a brand new part of the Java ecosystem
- To gain competitive advantage
  - Be the early experts in a new standard
- Gain more members
  - Helps build long lasting activities for the JUG
  - Turns passive members into active ones!
- It’s fun!

How does it help the standards?

- Standards get earlier feedback
  - Leading to more developer friendly APIs
- Standards get 'end user/developer' expert input
- Standards get developed more quickly, JUGs can help build
  - Reference Implementations (RI)
  - Technical Compatibility Kits (TCK)
- JUGs can also help with management of the JSR
  - Managing mailing lists
  - Triaging issues
  - Testing
  - Evangelism
  - + more!

Who’s involved already?

- See http://adoptajsr.org for up to date details.
  - The London Java Community (UK)
  - SouJava (Brazil)
  - GoJava (Brazil)
  - Houston JUG (USA)
  - Chennai JUG (India)

- + Several more starting out

What difference have they really made?

- SouJava helped push through JSR-348
  - Reforming the Java Community Process (JCP)
  - Open mailing lists, public Issue trackers
  - Standards reviewed early and often
- The LJC is helping make sure JSR-310 gets through
  - Date and Time for Java 8
  - Focussed on building the TCK
  - TCK ~40% complete as of Dec 2011
  - Ensures that we get the new Date and Time API for Java 8
- Houston JUG is working on Data Grids
  - Vital technology for the cloud space
  - Avoids vendor lock-in

How does my JSR join and get volunteers?

- Contact your local JUG leader
- OR
- Go to http://adoptajsr.org
  - Follow the simple contact information there

Lets change the Java ecosystem!

- Thanks for reading/listening!
  - The acronyms become familiar, we promise
  - Volunteers can be a benefit not a hindrance
- We really hope you'll consider using more volunteers
  - Duke would love their help
- http://adoptajsr.org for full details of the JUG lead program
- You can contact me on twitter:
  - Martijn Verburg (@karianna)