Eclipse Project Update

The Eclipse Project PMC
Strategy

• Seven releases in a row: innovation + stability every year
• Stability vs. innovation
• 3.x for consolidation
  ▶ stability, scalability,
  ▶ leverage newer platforms and Java versions,
  ▶ remain on JVM 1.4,
  ▶ controlled innovation
  ▶ 3.4 for Ganymede
• 4.0 for innovation
  ▶ increase participation to the platform
  ▶ work in e4 incubator
  ▶ 2 years time frame
• Equinox to move to RunTime (RT) top-level project
3.4 Release Plan

- [http://www.eclipse.org/eclipse/development/eclipse_project_plan_3_4.html](http://www.eclipse.org/eclipse/development/eclipse_project_plan_3_4.html)
- All planned items marked as committed or proposed. None marked as deferred
- Eclipse 3.4 will be compatible with Eclipse 3.3
- Focus on Platforms, Consumability, Reliability and The Future
- Highlights:
  - SWT 64-bit, SWT WPF port, Linux BiDi, Mac Carbon Internalization & accessibility, product level configurability, provisioning, serviceability, API tooling, security, concurrent compiler and Eclipse 4.0 planning
Self-assessment

- Openness and transparency
  - Better on PMC since Jeff left
  - Priority for development, in particular for e4
- Meritocracy
  - SWT: 2 new committers on SWT (1 IBM, 1 Intel), 1 more in progress (Adobe)
  - Equinox: 13 new committers (5 outside IBM)
  - Platform/UI: 2 new committers (1 outside IBM)
  - Need guidance on revoking commit rights from inactive committers
Self-assessment

- Diversity
  - Historically contributed by IBM. Not much improvement.
  - More committers outside IBM, but commits mostly from IBM, with help from BEA
    - Equinox: 89% IBM, 8% individuals, 1% ProsystSoftware, 1% compeopleAG
    - JDT: 95% IBM, 4% BEA, 1% individuals
    - PDE: 82% IBM, 17% individuals
    - Platform: 98% IBM, 1% BEA, 1% QNXSoftwareSystemsCo.
  - Essential to e4
    - 4.0 will only occur with help and participation from our community
    - Already some interest from RAP (Innoopract)
    - e4 incubator to recruit more committers on the platform
Self-assessment

- Compliance with the purposes
  - Best possible API design
  - Extensibility crucial to agnostic platform
  - Platform enhancements must be useful to community at large
    - Until you have several clients, you do not have an API

- End user community and adoption
  - Huge adoption today
  - How much adoption for Eclipse 4.0 vs Eclipse 3.x?
    - Cannot succeed with participation from multiple companies
Self-assessment

- Commercial community and adoption
  - Depended by all other projects
  - Thousands of Eclipse-based solutions across the world (350+ in IBM)
  - How quickly products adapt to newer versions of the platform?
    - New capabilities
    - Increased performance
    - Scalability, robustness
  - Still too often, platform evolutions mean breakage
    - Even if strictly compatible by our API charter
    - Use of internal interfaces means vulnerability to even a service release update
- 3.x / 4.0 split
- API tooling (3.4) to allow risk assessment with any component

- Compliance with the roadmap
  - Deliverables on schedule with planned content
  - In milestone-6 (API freeze); M7 is consolidation and performance
Board Assistance

- Improving diversity
  - Not just in 4.0, 3.x needs help as well
  - Update Manager at risk of having zero committers
  - Need more help from OS manufacturers for SWT ports
- Establishing Eclipse 4.0
  - The future of Eclipse is a big initiative
  - Need help to attract more committers and companies
  - Need to promote the technology for wider adoption
- Java Compliance Testing
  - JDT contributes a Java compiler
  - Foundation has no license to compliance tests (JCK)
    - Cannot assess compatibility as part of open-source dev
  - Regressions can be introduced with any change to the compiler
  - Need foundation to own a JCK license