Who are we?

Thiago Macieira

- Open Source developer for 15 years
- Software Architect at Intel’s Open Source Technology Center (OTC) and Tizen Platform Community Manager
- Maintainer of two modules in the Qt Project
- MBA and double degree in Engineering
- Previously, led the “Qt Open Governance” project

Guy Martin

- 20+ years in software development, open source consulting & community management
- Senior Strategist – Samsung Open Source Group (OSG)
- Developed open source & collaborative communities for Motorola, Sun, and US DoD
- Previously, built Red Hat's strategic open source consulting practice
Moving development to Tizen.org

- Moving Tizen from 'source-available' to truly open source / collaborative
- The main area of development and contributions is Tizen.org
- For productisation/commercialisation and depending on profile policies and rules, main code tree can be pulled and built anywhere by anyone
  - A small number of profile-specific packages may continue to be developed using the “old” model (shared private first, then moving to Tizen.org)
- Tizen.org is becoming the primary development hub for Tizen

Tizen.org is now the primary development hub for the Tizen Platform
Overall governance guidelines

- All community members **must**:  
  - Be respectful of one another  
  - Refrain from engaging in flame wars or personal attacks  
  - Ensure Tizen community rules are followed  
  - Follow general direction set by the TSG

- All community members **should**:  
  - Contribute to improve Tizen  
  - Help out new contributors and users  
  - Approach the TSG with important matters

Disrespect of the rules may be sanctioned
The principle of “lazy consensus” & silent consent

• **Procedure:**
  - It is enough to reach consensus in a small group
  - Those who disagree must provide alternative solutions
  - Those who did not participate silently and implicitly give their consent

• **Requirements:**
  - Public discussion, to the correct audience
  - Active membership that follows discussions
  - Reasonable time for objections
    - “Reasonable” is to be taken on case-by-case basis

Decision-making and meritocracy

• Responsibilities and rights are awarded to those who merit them
  – Merit gained by contributing to the project
• Decisions made at the lowest level
  – People with merit are allowed to make decisions
  – Lazy consensus applies, so:
    • Decisions are public
    • Other contributors are allowed to object
• Higher levels oversee decisions
  – If necessary, override
Project roles

- Developers
- Reviewers
- Maintainers*
- Architecture Workgroup

*: Exactly one maintainer per Git repository

- Release engineers
- QA engineers
Developer

• Role: develops code or other contributions

• Rights and duties:
  - Can participate in mailing lists
  - Can report bugs and suggestions
  - Can contribute code changes for review to appropriate branches
  - Can (and is encouraged to) participate in the review process (-1 and +1 reviews)
  - Should participate in relevant discussions in mailing lists
  - Should offer constructive advice to other developers
  - Should behave as a Reviewer in all aspects
Reviewer

- Role: makes decisions on contributions
- Rights and duties:
  - All of a Developer’s rights and duties
  - Can make decisions on code contributions (-2 and +2 reviews)
  - Should pro-actively participate in the review process
  - Should participate in the release process
  - **Must not** approve own contributions
  - **Must** offer constructive comments when rejecting a contribution
  - **Must** ensure contributions are ready for the production tree before approving
Maintainer & Integrator

• Role: maintains a package and ensures progress
  – Only one maintainer per Git repository

• Rights and duties
  – All of a Reviewer’s rights and duties
  – Can make more invasive changes to a Tizen package (e.g., create branches, rebase, etc.)
  – Can approve own contributions in exceptional cases (example: no other reviewer was available)
  – Interprets TSG medium- and long-term goals into short-term ones
  – **Must not** abuse maintenance rights to bypass the review process
  – **Must** ensure all contributions to the maintained package are reviewed
  – **Must** release approved contributions into the product buildsystem
  – **Must** ensure quality of the package
  – **Must** participate in the Tizen quality and release process when required
Other roles

Release engineer

• Role: create pre-release and release images

• Rights and duties:
  – All of a Developer’s rights and duties
  – Should report bugs and regressions
  – **Must** manage Tizen’s build system
  – **Must** approve or reject submissions to the build system
  – **Must** create Tizen images and smoke-test them
  – **Must** participate in the release process

QA engineer

• Role: ensure the overall quality of Tizen releases

• Rights and duties:
  – All of a Developer’s rights and duties
  – **Must** test Tizen images
  – **Must** report bugs and regressions
  – **Must** participate in the release verification process
Becoming Reviewer, Maintainer, etc.

- Requirements: candidate is **already** acting at that level
- Procedure:
  - Nomination by an existing project member of that level
  - Agreement by another one (all levels) or two (maintainer) other members
  - Motion carries if no objections are raised within three weeks
- Specifically for maintainers:
  - There is a package in need of a Maintainer
  - Candidate is already a reviewer in some part of Tizen
Architect (member of the TSG Architecture Workgroup)

• Role: oversees and advises in technical matters
  – Decisions are always of the entire Architecture Workgroup

• Rights and duties:
  – All of a Reviewer’s rights
    • Some architects may be Maintainers in their own right
  – Should pro-actively offer advice to on-going discussions
  – Interprets long-term Tizen strategy and vision into long- and medium-term technical goals
  – **Must** monitor the overall health and progress of Tizen
  – **Must** participate in Architecture Workgroup decisions (lazy consensus applies)
The TSG Architecture Workgroup

• Members:
  – Group of senior engineers or architects (experts) with broad view of Tizen
  – Initially, 5 to 8 members
  – Can be expanded later, by self-selection

• Decision-making:
  – Periodic meeting, all architects plus invited experts
  – Decisions posted publicly
  – Lazy consensus: architects who do not object need not speak up
Sanctions / Revocation of rights

• All contributors, at all levels, **must** obey community rules

• In case of **continued** disrespect, the community may impose sanctions
  - We need to be understanding of the varied cultural backgrounds
  - Community Management is available to help mediating issues

• Cases will be dealt with on a **case-by-case** basis, in a closed Architecture Workgroup & Community Management meeting
Principles of the Tizen codebase

- Keep code as close to upstream as possible
- Upgrades must be justified
- Goal is to have 1 project run on multiple profiles
  - Same code base but with different build options and enabled features
  - Contributors have to focus on the whole project rather than a specific profile
  - Avoid hardcoding
Flow of a contribution

Developer
- Write code
- Submit to Gerrit

Reviewer
- Offer suggestions
- Review
- Not ok
- Ok
- Approve contribution

Maintainer
- Release to buildsystem (multiple changes)
- Integrate contribution

Release engineer
- Not ok
- Create image
- Smoke-test
- Ok
- Release image
What tools we use

• Tools used:
  - Gerrit, for code and contribution reviews
  - JIRA, for reporting bugs and keeping track of tasks
  - Email and mailing lists, for discussions
Mailing list for discussion

- Platform development uses one main mailing list:
  - dev@lists.tizen.org
  - All project-wide discussions should be on this list
  - TSG Architecture Workgroup decisions are posted to this list
  - In the future, we may split if traffic becomes too high

- Members and rules:
  - Open to subscription and posting from anyone on the Internet
  - Decisions respect lazy consensus and meritocracy
  - Common-sense and regular “netiquette” apply
    - List moderators will act in case of abuse
  - No confidential material: all discussions are public
Bug reports and task tracking (JIRA)

- Manages defects from all public binaries
- Contains the official release criteria
Code reviews (Gerrit)

- **Day-to-day decisions:**
  - Comments, enhancements
  - Approvals

- **People:**
  - Everyone can create an account
  - Everyone can participate
  - Everyone can offer +1 or -1
  - Reviewers can approve (+2) or permanently reject (-2)
  - **Only** Maintainers can integrate the change
Code contribution approval guidelines

Technical / Objective rules

- Does this compile?
- Does it introduce a bug?
- Does it do what it is says it does?
- Does it follow the coding guidelines*?
- Does it follow the Tizen branch guidelines?

Subjective rules

- Is this in the direction set forth by TSG, Arch WG, Maintainer?
- Does it introduce risks to stability?
- Is this the best solution?
- Is this the right time?

*: coding standards, defensive coding, file naming conventions, etc.
# Decision-making

<table>
<thead>
<tr>
<th>Type</th>
<th>Who*</th>
<th>Where</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code contribution</td>
<td>Reviewers of the package</td>
<td>Code review tool</td>
</tr>
<tr>
<td>Decisions on one package</td>
<td></td>
<td>Mailing lists</td>
</tr>
<tr>
<td>A few packages</td>
<td>Reviewers and Maintainers of the packages</td>
<td>Mailing lists</td>
</tr>
<tr>
<td>Small / low-impact issues</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A lot of packages</td>
<td>Reviewers, Maintainers and Architects</td>
<td>Mailing Lists TSG Architecture Workgroup meeting</td>
</tr>
<tr>
<td>Large / big-impact issues</td>
<td></td>
<td></td>
</tr>
<tr>
<td>API in the Tizen Compliance</td>
<td>Reviewers, Maintainers and Architects</td>
<td>Mailing Lists TSG Architecture Workgroup meeting</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*: people typically involved. It's always possible for more people to participate.
Branching plan drawing

- Tizen 3.0 release
  - Tizen 3.0.1

- Tizen 3.1

- New features & bugfixes
- New features & bugfixes
- Bugfixes

New features & bugfixes

Copyright © 2013 Intel & Samsung. Public.
Conclusion

• Tizen’s Open Governance is live
• Model by which we work with each other and partners
• Suggestions to improve are welcome
• Join dev@lists.tizen.org to participate
  – Only requirement: your will to improve Tizen!
  – Be nice and professional
Thank You!

Thiago Macieira
thiago.macieira@intel.com
Guy Martin
guy.martin@samsung.com

dev@lists.tizen.org
http://source.tizen.org