Can Peer-Code Reviews be Exploited for Later Information Needs?

Andrew Sutherland [U. Saskatchewan] and Gina Venolia [Microsoft Research] – HIPS Human Interactions in Programming group

Field Study of Current Practice

Methodology
- Formal survey (157 SDEs, 134 SDET)
- Interviews
- Email thread analysis
- Prototyping and feedback

Observations
- Almost all checked-in code is reviewed
  - All or almost all: 65%
  - About 75%: 11%
  - About 50%: 7%
  - About 25%: 10%
  - None or almost none: 6%

  Equal reliance on async and sync modes
  - Diff packs and email
  - Over the shoulder
  - Could benefit from workflow tool
    - Email: 83%
    - Face-to-face: 75%
    - IM: 31%
    - Check-in comments: 16%
    - Phone: 16%
    - Source-code comments: 13%
    - Workflow Tool: 6%

- But it’s not systematically retained
  - 31% of async reviews are retained for reviewers or team
  - 17% of sync reviews are retained for reviewers or team

- Desire to use knowledge captured during review later
  - 63% would be “likely” to use a tool for this

- Many reasons to refer later to reviews
  - Understand change rationale: 54%
  - Track changes to code: 45%
  - Verify bug fixes: 43%
  - Future review preparation: 35%
  - Maintain awareness of team: 30%
  - Other: 6%

Guidelines for Tool Support

- Tools should support the existing author-reviewer workflow
- Views of the code both prior to and after the change should be available
- Difference lists should be organized logically – by tasks or groups of changes
- Reviewers and authors need the ability to include additional information – such as comments or notes
- Review data needs to be retained in a systematic way
- Past review data needs to be recoverable so that authors and reviewers can utilize this information later

Crosscheck Tool: Support & capture reviews

- Author identifies tasks, tags diff-blocks, and assigns reviewers
- Reviewers view diffs by task; comments on, accepts or rejects diff-blocks or whole tasks
- Author makes changes, answers comments; the review cycle continues
- Async or over-the-shoulder
- Knowledge is retained in a database (search, browse and link)

Future Work

- Querying and visualization of review data
- Automation for initial task identification
- Structural diff
- Integration into modern development environments
- Better support for synchronous code reviews with video and audio recording