An Active Help System to Improve Program Navigation
Petcharat Viriyakattiyaporn and Gail C. Murphy
Software Practices Lab, University of British Columbia, Vancouver, Canada

Problem
- Development environments (IDEs) include many tools to aid software developers.
- Developers often do not know about all tools available.

Spyglass UI
- We carefully designed Spyglass’ UI so that it presents its recommendations politely.
- The Spyglass icon, which is the entry point to all recommendations, is located in the bottom-right corner of the main IDE window.

Proposed Solution: Spyglass
- Our Spyglass recommender system suggests appropriate navigational tools within IBM’s Rationale Team Concert IDE.
- Spyglass infers navigation between Java code elements, work assignments (work items), and sets of program revisions (changesets).

Lab User Study
Participants
- 18 participants with
  - 8 months of Java
  - 2 months of Eclipse.
- Completed two training tasks on Paint program.
- Divided participants into a tutorial group and a Spyglass group.
- Independent-measures design.

Spyglass Group
- Read the given Spyglass introduction.
- Completed two change tasks on JFreeChart program (with Spyglass UI disabled).
- Filled out a post-test questionnaire and had interview.

Tutorial Group
- Read the given passive tutorial document (excluding the target tools).
- Completed two training tasks on Paint program.
- Completed two change tasks on JFreeChart program (with Spyglass UI enabled).
- Filled out a post-test questionnaire and had interview.

Results
1. Can Spyglass help developers complete tasks more successfully?
2. Can Spyglass help navigate through pieces of information more efficiently?
3. Spyglass group were aware of and used the target tools more?
   - As much as the passive tutorial.
4. Developers felt the recommended tools were useful?
   - Yes for Open Call Hierarchy; Neutral for Open Type Hierarchy.
5. Developers felt Spyglass suggested the right tool at the right time?
6. How often does Spyglass suggest the right tool at the right time?
   - Half of the time (precision ~ 0.53).
7. Spyglass is not disruptive?

Conclusion & Future Work
- Spyglass needs to provide more meaningful and accurate recommendations.
- May be achievable with a more complete user model, finding the best configuration parameters for the system, and revising our algorithm.
- Explore having Spyglass recommend other types of tools.
- Thorough comparative studies between existing approaches on active help systems and our approach in a software development environment are needed.