Putting Examples to Work

Reference Implementations for High-Volume Automated Testing

Carol Oliver

Workshop on Teaching Software Testing 2013

Why Reference Implementations

- Glacially slow adoption even of known techniques
- Implementation from nothing is intimidating
  - Conceptually overwhelming, rendering people to consider technique descriptions as too theoretical to apply
  - Technically overwhelming, requiring more technical skills or time than available
- What if we give people working examples with code they can modify?

Our Commitments

- Working examples
- Maximum practical separation between domain-specific details and core code that does the technique-specific heavy lifting
- Quality reference code to guide custom development
- Streamlined, well-explained concrete example to model after
- Robust unit test support for the tool
- Additional supporting materials where sensible and practical

General Design

- One Platform (Window 7, Ruby 1.9)
- Individual Tools, each configurable to apply to a variety of test domains (some programming required to do so)
- Common Codebase across tools where intelligent to have it
- Freedom for specific tools to use specialized code if that’s what makes the most sense

Open Questions

- What is the quality standard for a reference implementation?
- What level of generality do we want to see in the architecture?
- What is the minimum level of sophistication we expect adopters (e.g. testing staff) to have?

Open Questions – 2

- What uses will this reference implementation have?
  - Adoption at Ford to test embedded car software?
  - Adoption in courses?
  - Self-study?
  - Adoption by other research students?
- What barriers to adoption?
  - Technological
  - Conceptual

WHAT DETAILS ENHANCE ADOPTABILITY?

Grace

- Currently an administrative assistant
- Wants to move into a higher-paying technical job
- Helped test software on a volunteer basis and now is getting formally assigned a few hours at a time to some testing projects
- Bright, curious, not afraid to try things
- Highly motivated to bring value; wants to be a contender for the next test position to be funded

Grace Finds a HiVAT Tool

- Needs a clear description of what the tool can do
- Wants a clear description of some ways the tool could be applied
- Wants to be able to try it out immediately, to see if it’s worth further exploration time

Grace Installs the HiVAT Tool

- Might not be able to install it on her work computer, if her company strictly regulates who can install software and what software can be installed on most computers
- Willing to try it on her personal computer
- Needs the install to be simple and clean
  - One or only a few components to install
  - Clear installation documentation
  - Clear uninstallation documentation
  - Clean and complete uninstall, no lingering side-effects

Grace Tries the HiVAT Tool

- Needs clear instructions how to set up the tool to do something, how to run it, how to work with the results
- Wants the tool to work out-of-the-box with software already on her computer (e.g. Windows accessory programs) or with tiny demo programs bundled and installed along with the tool
- Wants instructions for how to tweak the demo configuration to alter the default functionality (to show how modifying a fully-set-up instance of the tool can work)

Cecily

- Full-time tester
- Programming experience is all in her past, not used routinely in her present job
- Not familiar with Ruby
- Lots of local domain expertise
- Tasked to trial the tool, evaluate whether it really helps the group do a better job and to see if test group can use it without distracting programmers

Cecily Tries the HiVAT Tool

- Needs clear instructions how to alter the tool to apply it to a different test domain
- Needs as limited as possible a scope of new code to write and existing code to alter
- Needs model code to imitate
- Needs instructions and examples of how to extend testing scope within chosen domain using this technique’s approach

Helping Cecily

- Probably benefits from a walkthrough tutorial against a reference program
  - Bridge into tool customization
  - Bridge into Ruby
- Probably benefits from teaching commentary in the code rather than being dropped into an expectation of expertise

Coaching Cecily

- Probably benefits from coaching on moderately sophisticated design and maintenance of test code and test suites
  - Breaking down test problems into reusable components?
  - Advice for how customizations will be more easily maintained and extended over time?

Refining the Plan

- Who else?
  - What needs do they have?
  - What wants do they have?
  - What would they benefit from which they may not know to wish for?

Reprise: Open Questions

- What uses will this reference implementation have?
  - Adoption at Ford to test embedded car software?
  - Adoption in courses?
  - Self-study?
  - Adoption by other research students?
- What is the minimum level of sophistication we expect adopters (e.g. testing staff) to have?
- What details enhance adoptability?

Reprise: Open Questions – 2

- What is the quality standard for a reference implementation?
- What level of generality do we want to see in the architecture?
- What barriers to adoption?
  - Technological
  - Conceptual
- What details enhance adoptability?