

AnnoTestWeb/Run: Annotations based Acceptance Testing

David Connolly, Frank Keenan and Fergal Mc Caffery

Regulated Software Research Group
Software Technology Research Centre
Dundalk Institute of Technology, Ireland

Introduction

Acceptance Testing

- “Data supplied by customer” [1]
- End stage
- Expensive phase to detect errors

Agile adaptations to Acceptance Testing

- Short iterations means ATs part of process from start
- Acceptance Testing synonymous with functional testing [2]
- Executable artefacts make ATDD possible

Existing Tools

- FIT [3]
- FitNesse [4]
- Greenpepper [5]
- FITclipse [6]

Challenge

- Support identification of potential acceptance tests by annotation of existing documentation.

Why Annotations?

- Annotation format evaluated experimentally by authors
- We found annotations assisted in the identification of tests

Approach

- Import existing documentation, review and annotate elements of acceptance test
- In absence of existing documentation, write narrative descriptions of test in an agile team environment and then annotation.

Tool Support

- Easily and quickly identify elements of acceptance tests in text
- Support simple grouping of tests

Annotations

- Based on elements of good acceptance tests [7]
- Annotation types for three elements:

- Act (Based on Actors & Actions)
- Data (Based on Examples)
- Result (Based on Observables)

N.B. The fourth element “pre-conditions” are implemented using order of tests in a document

- Annotations have some additional meta-data to support model:

- to label elements of the same type (for repetitions)
- to describe data types
- change default value (other than annotated text)

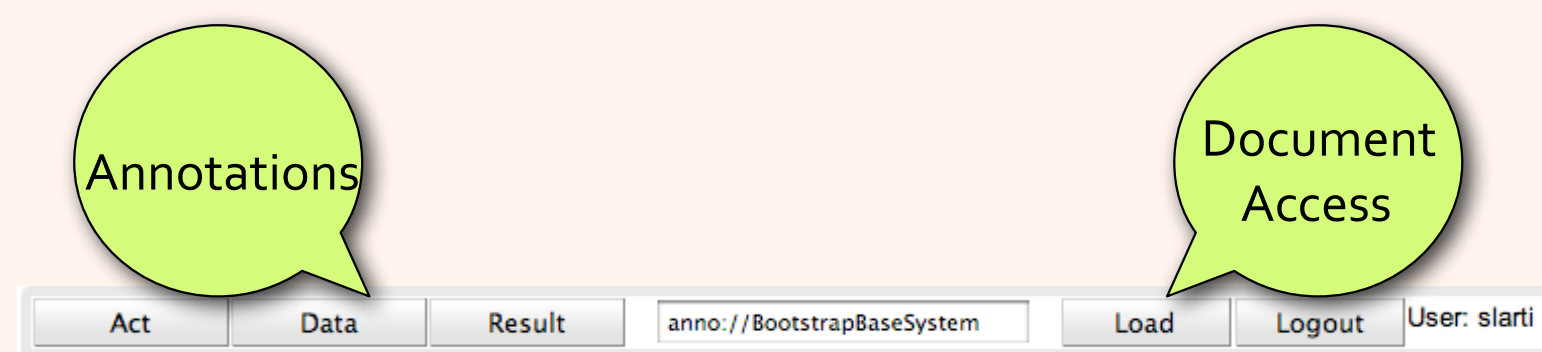
Technical Overview

Components

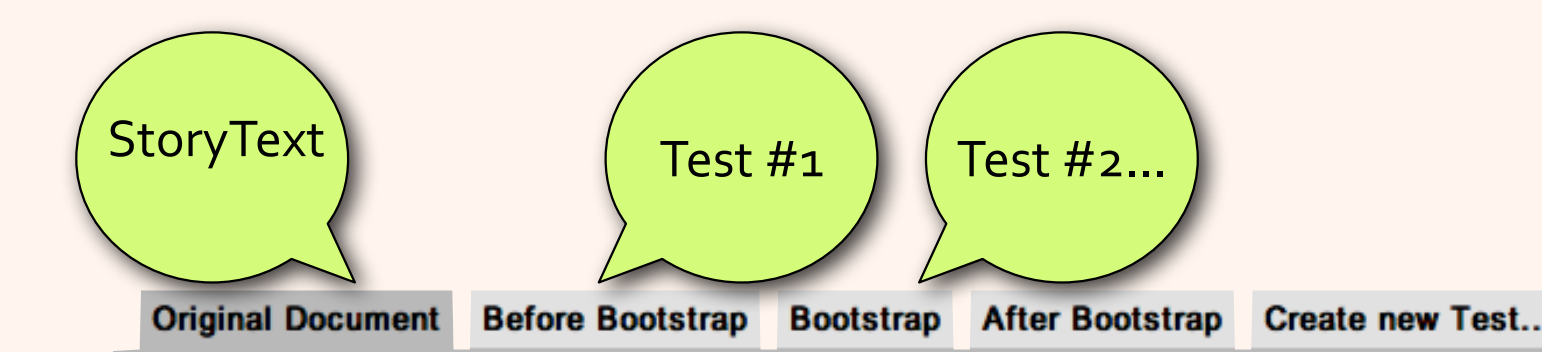
- GWT, CouchDB
- Open source allowed development to focus on features!

Interface Elements

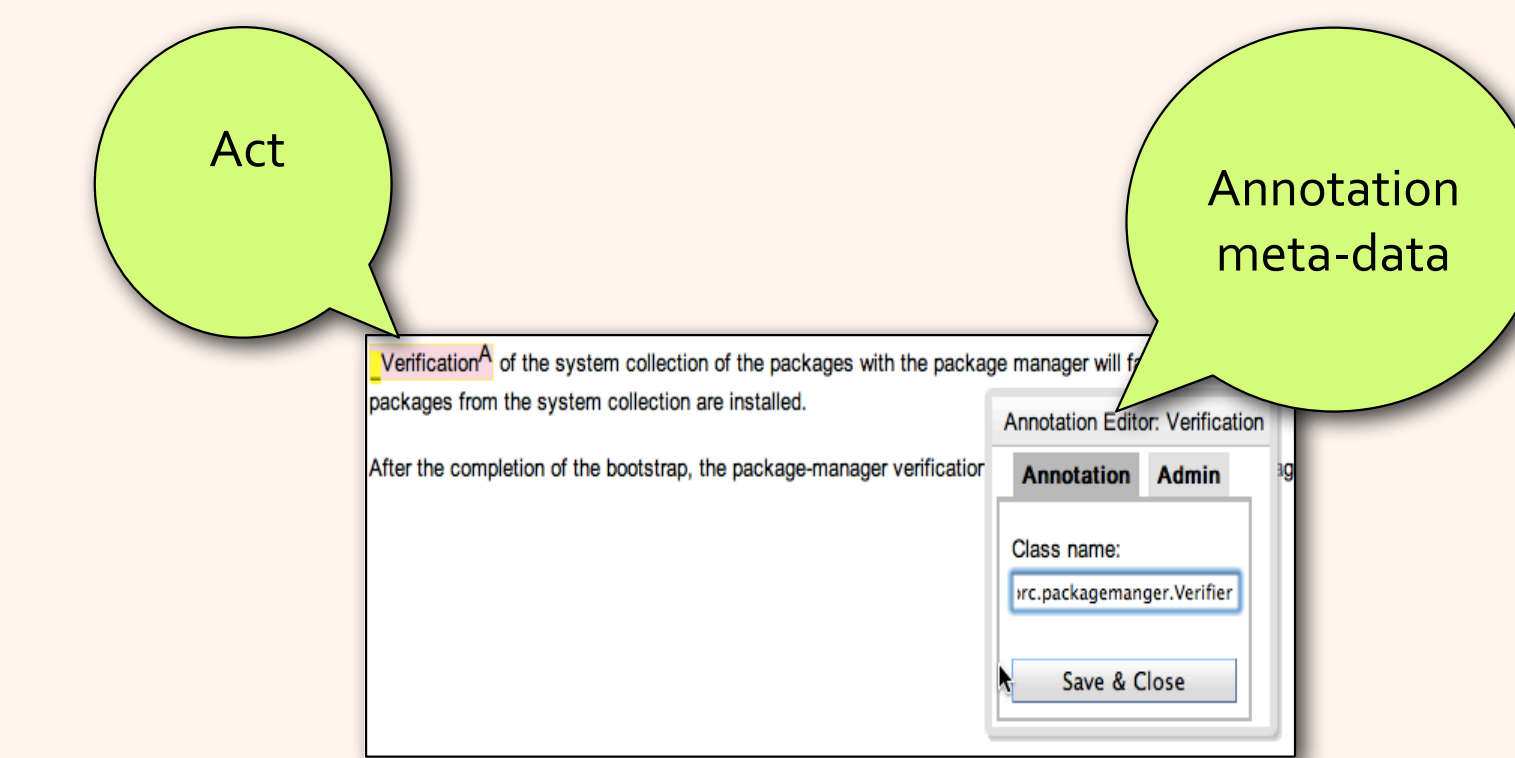
- Creating annotations and managing documents



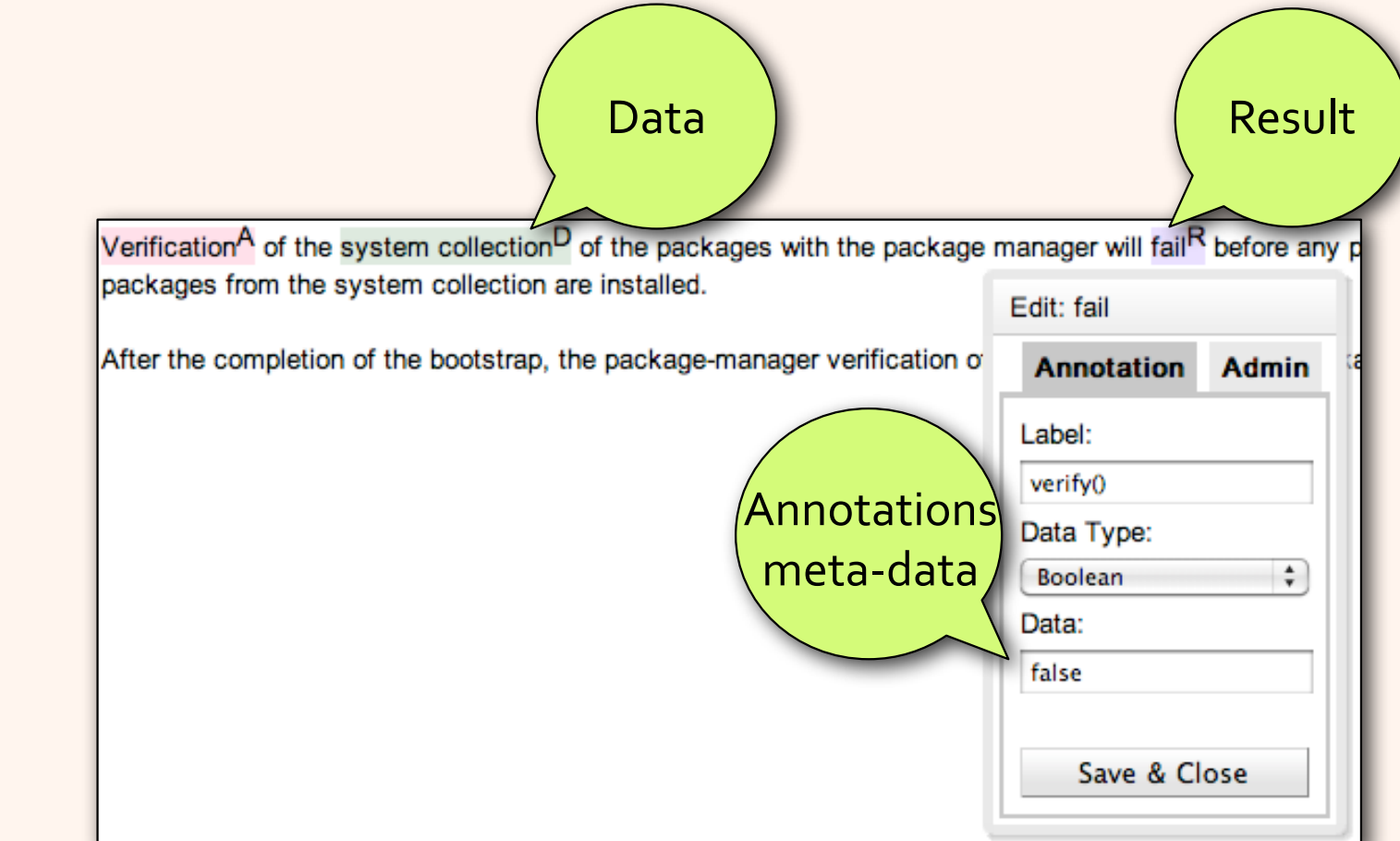
- Managing and creating tests within document (e.g. Test #1 is a precondition of Test #2)



- Creating test by adding annotations to a document



- Specifying detail by adding metadata to annotation



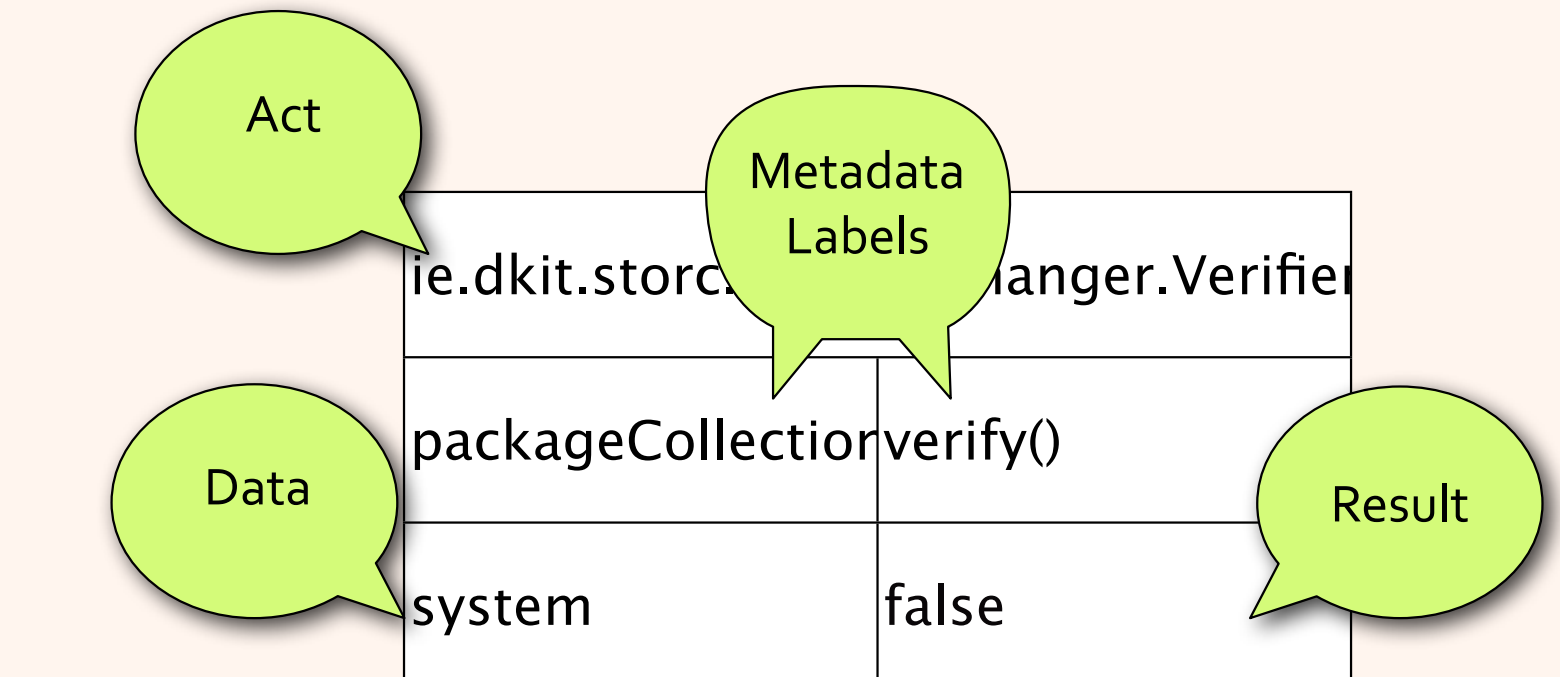
Test Reports

- Lightweight Fixture Code:
 - Fixture consumes data in test
- Javascript Object Notation (JSON) reporting mechanism
 - Reporting expected vs. observed results of test
- JSON Reporting Mechanism API ported to:
 - Java
 - GWT
 - C#

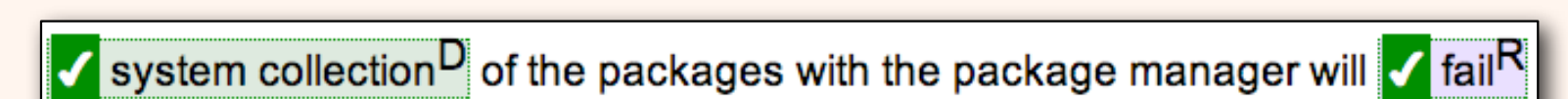
Test Reporting Style

Model of acceptance test captured by annotating document combined with reporting multiple reporting formats.

- FIT Table style report of test run



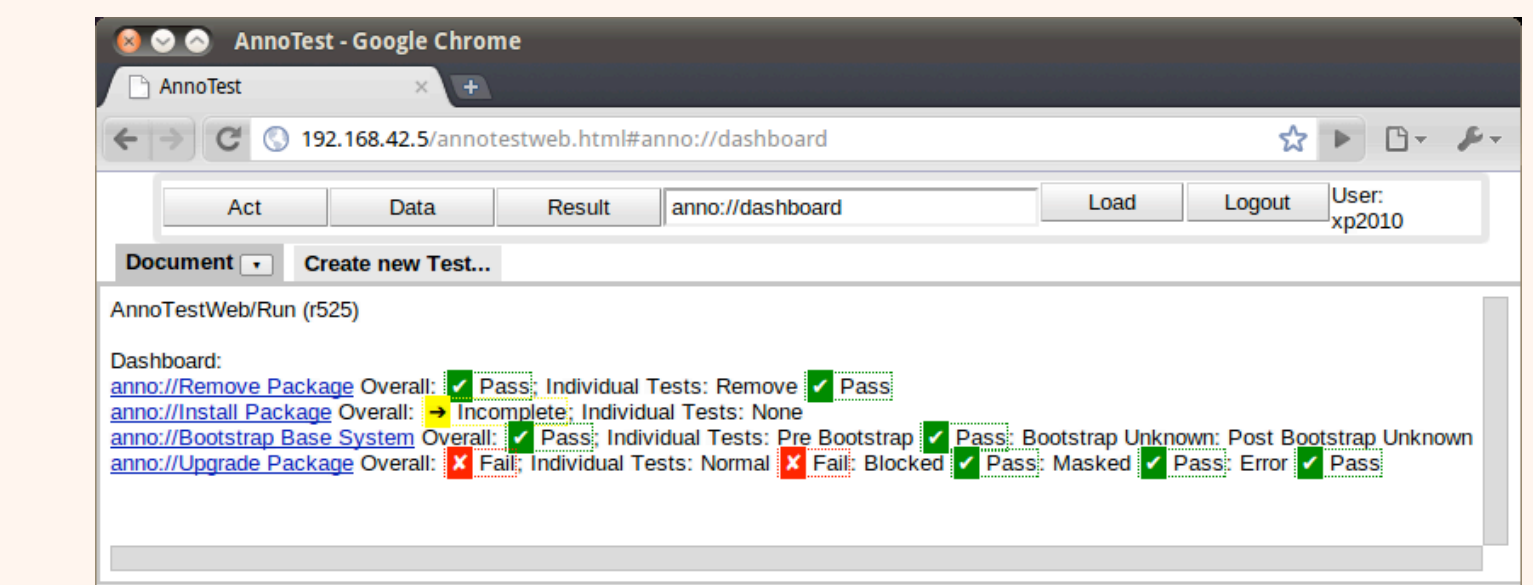
- Annotation based report of test run



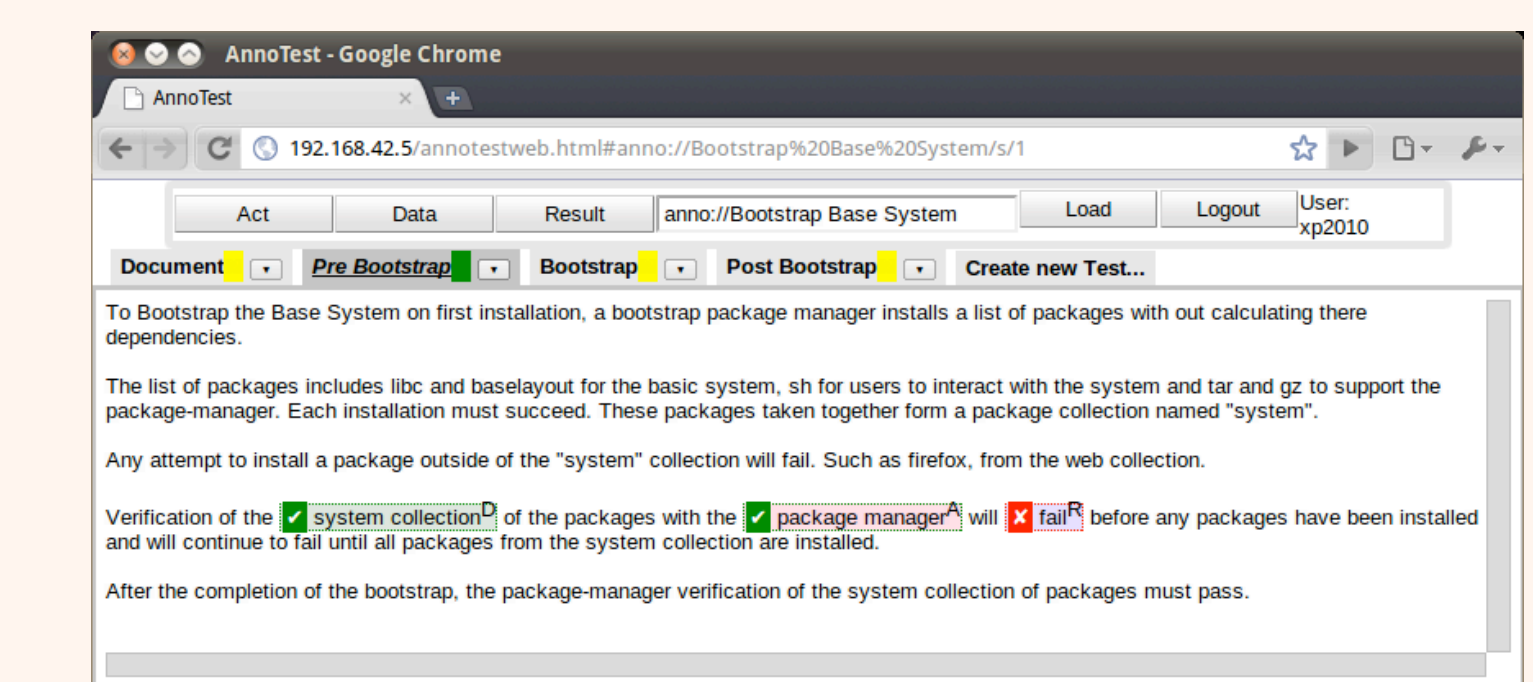
- Hybrid approaches and tracking test runs across time/iterations.

Usage Screenshots

- Dashboard containing Test Documents including status of most recent test executions:



- Manual Test Execution UI (without JSON Notification API or when a manual step exists in test)



Video Demonstration of Usage Scenario

- A five minute demonstration of prototype tool (YouTube)

<http://bit.ly/8XDnoc>



- Be sure to have your say by completing the short survey after the video demonstration!

Evaluations and Future Work

- Conducted demonstrations of tool with:
 - A domain expert, concerning document reuse
 - An agile team, currently working with ATDD
- Empirical investigation of tool:
 - Prototype tool usability experiments with undergraduate and postgrad students
 - Additional experiments
- Case studies of longer term use:
 - Considering use by an agile team
 - Focused on document reuse (In Progress)
- Evaluations used to inform future changes to the tool
- Future open source release of the tool

References

- Sommerville, I. Software Engineering, 8th edition, pages 80-81, Addison-Wesley, 2007.
- Sauv'e, J. P. and Neto, O. L. A. Teaching software development with ATDD and EasyAccept. In SIGCSE '08: Proceedings of the 39th SIGCSE technical symposium on Computer Science Education, 2008, pages 542-546
- Cunningham, W. FIT: Framework for Integrated Testing. <http://fit.c2.org>
- FitNesse. <http://fitnesse.org>
- GreenPepper. <http://www.greenpeppersoftware.com>
- C. Deng, P. Wilson, and F. Maurer, "Fitclipse: A fit-based eclipse plug-in for executable acceptance test driven development," in Proceedings of (XP 2007) (Springer, ed.), 2007.
- N. Jain, "Acceptance Test Driven Development". <http://www.slideshare.net/nashjain/acceptance-test-driven-development-350264/>

Acknowledgements

This research is partially supported by Institutes of Technology, Technological Sector Research Programme, Strand 1 Fund and Science Foundation Ireland through the Stokes Lectureship Programme, grant number 07/SK/1299.



For further information contact:
david.connolly@dkit.ie