

Accumulator

More Than a Simple Feed-Reader

Oliver Xylander, Johannes Wollert, Thomas Stoff, Konstantin Haase, Tim Felgentreff

Softwaretechnik I
Hasso-Plattner-Institut
Universität Potsdam
SS 2009

Introduction

- Common Feed-Readers have shortcomings
 - RSS-Feeds are often broken
 - Readers have limited input options
 - Feeds are only presented as they are



Introduction

- Common Feed-Readers have shortcomings
 - RSS-Feeds are often broken
 - Readers have limited input options
 - Feeds are only presented as they are
- The Accumulator is meant to solve some of these
 - Filters can fix some feed's problems
 - We can read more than the common feed formats
 - Feeds can be tagged and shared



Demo

Qwertial Aphasie

Add Feed
Edit User
Show All Items
Logout

xkcd
comics

Show all Show unread

Qwertial Aphasie

Edit Feed Remove Feed

I HATE HOW WHEN I'M TALKING WHILE I TYPE, SOMETIMES I ACCIDENTALLY TYPE A WORD I'M SAYING.

WANNA GO GET FOOD LATER?

CHECK OUT WHAT I FOUND IN THE CLOSET!

SORRY, I REALLY SHOULDN'T.

AWW, WHAT AN ADORABLE STUFFED GIRAFFE!

I CAN'T AFFORD TO KEEP EATING OUT THIS GIRAFFE.

FREQUENTLY! I MEANT "FREQUENTLY"!

2009-07-01T04:00:00+00:00

Idiocracy

Edit Feed Remove Feed

http://xkcd.com/

2009-06-29T04:00:00+00:00

Overstimulated

Edit Feed Remove Feed

http://xkcd.com/

2009-06-26T04:00:00+00:00

Game Theory

Edit Feed Remove Feed

http://xkcd.com/

Agenda

- Goals
- Development Process
 - Iterations
 - Xtreme Programming
- System Overview
- Requirements
- Conclusions



Goals

- Read from different sources (RSS, Atom, IMAP, Monticello, Git, ...)
- Filter sources to ease focus on the things that matter
- Be extensible: allow easy integration of new sources and filters
- Be open: do not force users into the web-frontend, offer RSS output



Development

- Xtreme Programming

- Planning game

- Pair programming

- Small releases

- On-site customer

- Test-Driven-Development

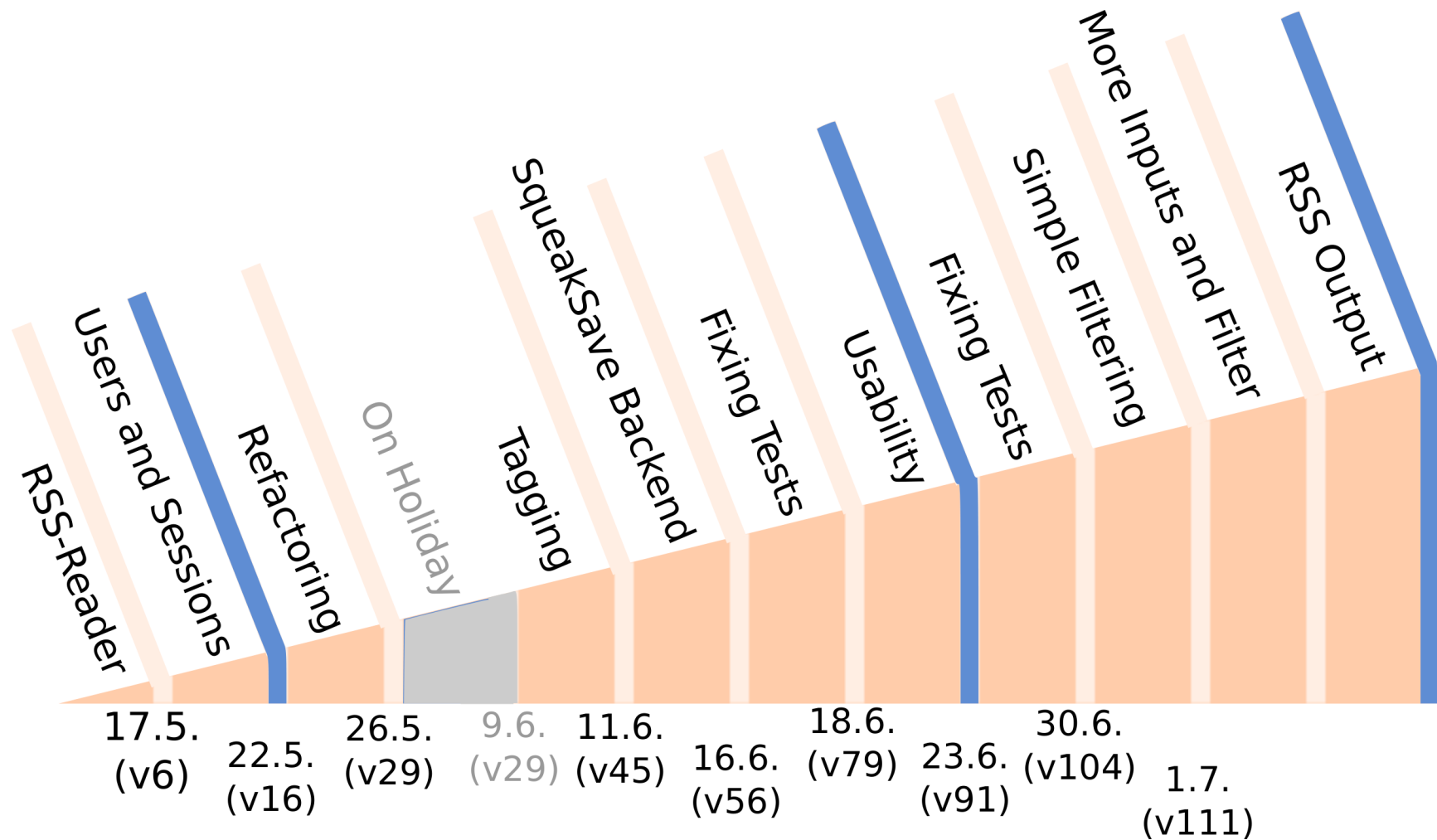
- Refactoring

- Coding Standards

- Simple Design, Metaphor

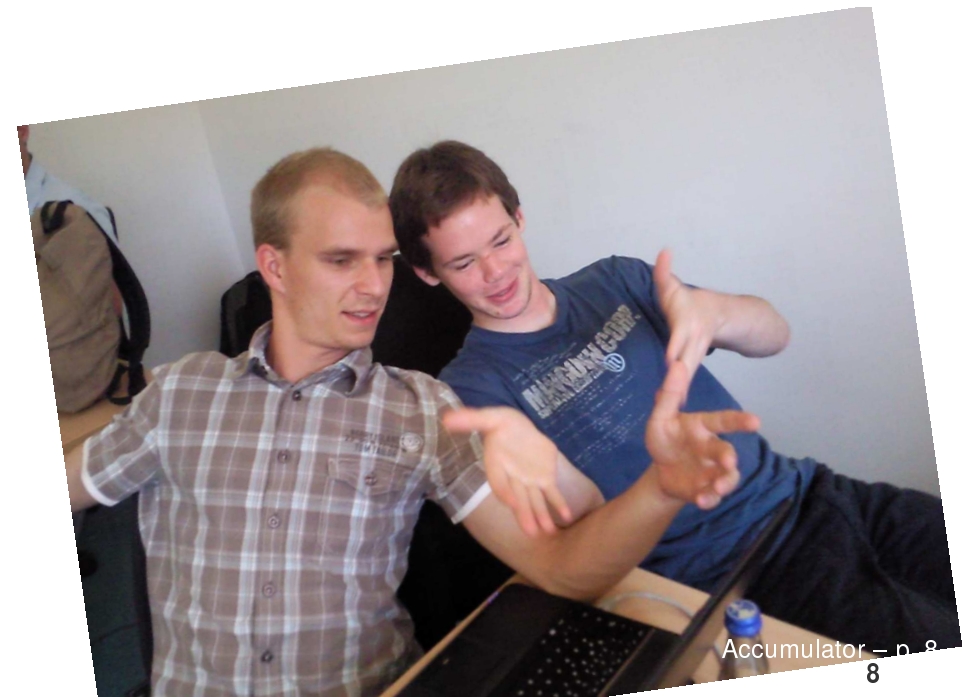


Iterations and Milestones



Development Process (contd.)

- Pair Programming
 - Working in two's or three's
 - Two or three teams working alongside on different parts of the system
 - Working in one room, each team on two or more monitors
 - “Development by smoke-break”



Development Process (contd.)

- Planning Game and Small Releases
 - Once every week
 - Bring everyone up-to-date with the latest changes
 - Discussion of last-weeks changes
 - Assignment of user stories to new pairs

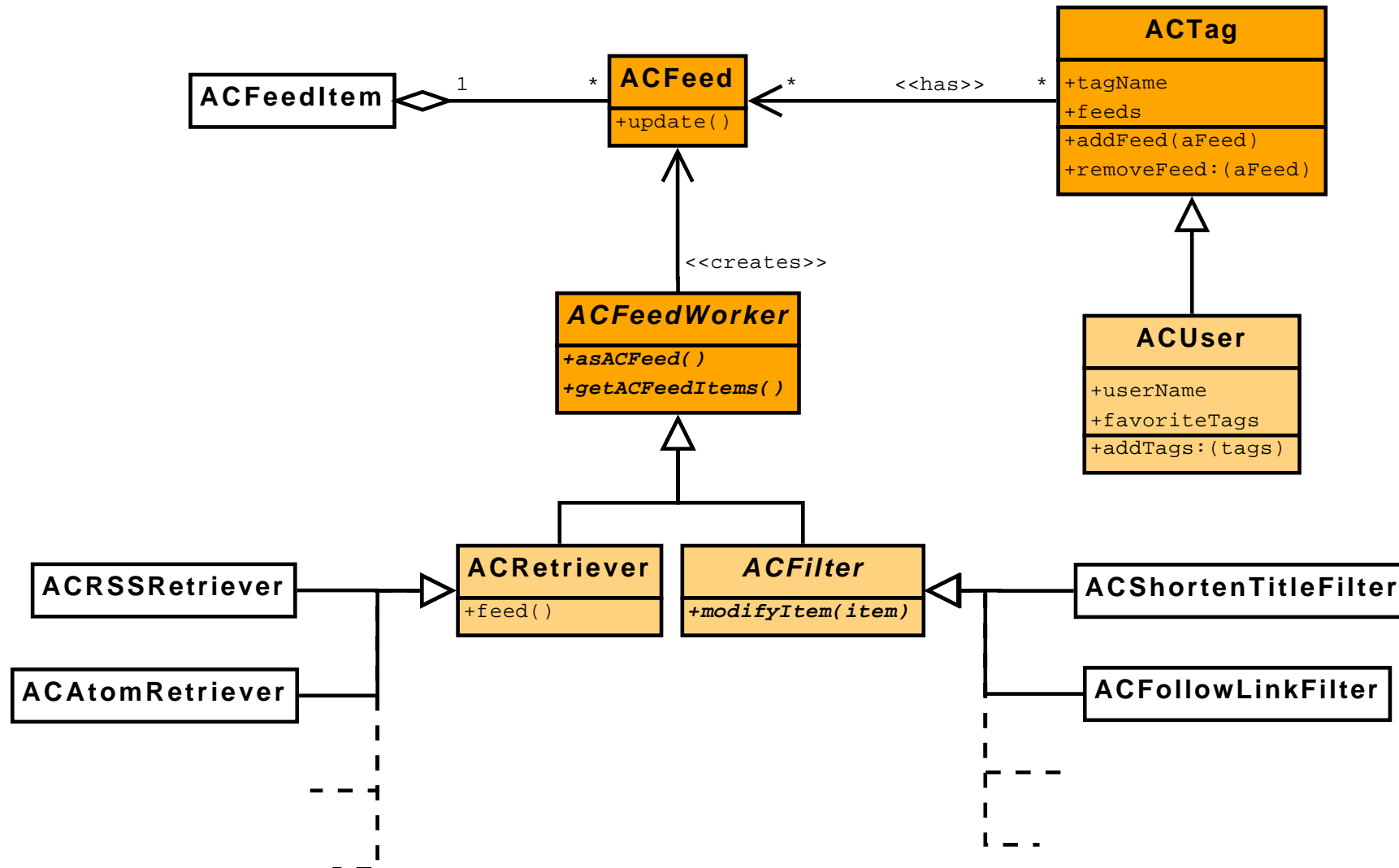


-
- Run Tests
- All Standard Tests
 -- filters (select) --
 -- filters (reject) --
 To Do
 Expected Failures
 Not This Version Or Platform
- KernelTests-Chronology
 KernelTests-Classes
 KernelTests-Methods
 KernelTests-Numbers
 KernelTests-Objects
 KernelTests-Processes
 KernelTests-Contexts
 CollectionsTests-Abstract
 CollectionsTests-Arrayed
 CollectionsTests-Sequence
 CollectionsTests-SkipLists
 CollectionsTests-Stack
 CollectionsTests-Streams
 CollectionsTests-Support
 CollectionsTests-Text
 CollectionsTests-Unordered
 CollectionsTests-Weak
 Exceptions-Tests
 Files-Tests
 GraphicsTests-Primitives
 GraphicsTests-Text
 GraphicsTests-Files
 MorphicTests-Basic
 MorphicTests-Kernel
 MorphicTests-Text Support
- TestCase
- ACTestHelper
 ACControllerTest
 ACApplicationController
 ACItemControllerTest
 ACLoginControllerTest
 ACTagControllerTest
 ACDatabaseTest
 ACFeedItemTest
 ACFeedTest
 ACFeedWorkerTest
 ACFilterTest
 ACFollowLinkFilterTest
 ACSimpleFilterTest
 ACRetrieverTest
 ACAtomRetrieverTest
 ACRSSRetrieverTest
 ACSessionTest
 ACTagTest
 ACUserTest
 ACViewTest
 ACItemViewTest
 ACLoginViewTest
 ACTagViewTest
- 47 run, 41 passes, 6 failures, 0 errors,
- ACFeedTest>>#testCreate {tests}
 ACItemControllerTest>>#testAppendTag {tests}
 ACItemControllerTest>>#testRemoveTag {tests}
 ACItemControllerTest>>#testShowTag {tests}
 ACItemControllerTest>>#testShowTags {tests}

Development Process (contd.)

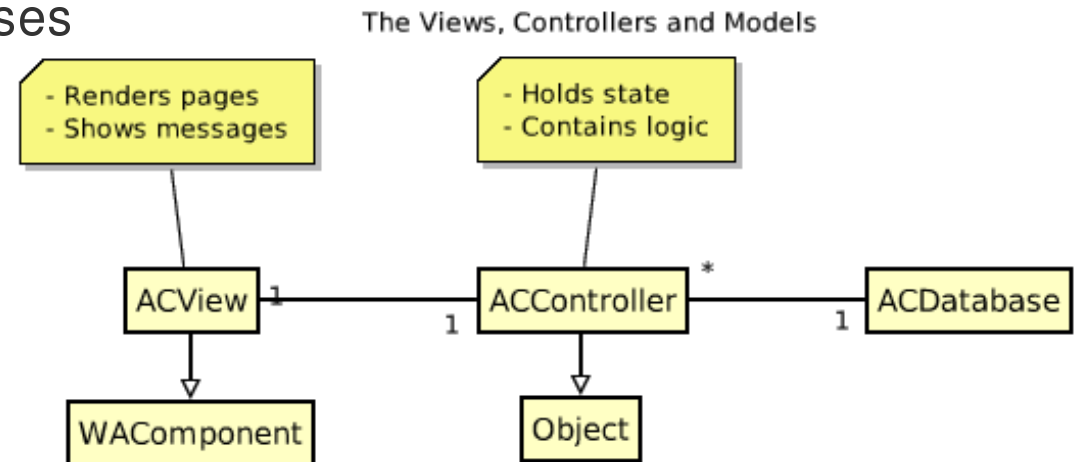
- Coding Standards and Refactoring
 - Always have a running version
 - After each change: run the test suite
 - Each commit should (within reason) be usable
 - Non-local commits should always add a feature or fix a bug
 - Breaking commits should be local

System Overview



Working the Code

- Convention over Configuration
 - MVC Pattern
 - Convenience method aliases
- Magritte
- Testing with mutations



Requirement Tracing: Filters

“Carl-Maria is pretty annoyed with Fefe’s RSS-Feed. He would like to have it’s title properly displayed in the content area, and at the same time shorten the title to only show the beginning of the first line. To do this, he adds two simple filters to the feed”

Requirement Tracing: Filters

“Carl-Maria is pretty annoyed with Fefe’s RSS-Feed. He would like to have it’s title properly displayed in the content area, and at the same time shorten the title to only show the beginning of the first line. To do this, he adds two simple filters to the feed”

- Offer a set of simple, well-defined filters
- Allow chaining of those filters to offer powerful filtering
- Create test-cases as contract for the I/O requirements
- Write the filter functionality and make it pass the tests
- Expose the filter through the frontend

Requirement Tracing: Tagging

“Johannes feels, directory hierarchies do not account for the diversity of information available in feeds. He’d much rather assign topics to feeds and be able to select feeds by topic. If at it, he would also love to be able to read what other people read on this topic”

Requirement Tracing: Tagging

“Johannes feels, directory hierarchies do not account for the diversity of information available in feeds. He’d much rather assign topics to feeds and be able to select feeds by topic. If at it, he would also love to be able to read what other people read on this topic”

- Use tagging instead of directories
- Feeds can have arbitrary tags - by selecting tags, the user gets a selection of feeds which belong to all selected tags
- A user is just a tag himself - when removing the user from the tag selections, tags from other users can be shown

Prospect

- Asynchronous updates
 - Threaded updating of the feeds in the server
 - Controller check for new Tags and Feed-Items regularly
 - AJAX for the views to show new information

Prospect

- Asynchronous updates
 - Threaded updating of the feeds in the server
 - Controller check for new Tags and Feed-Items regularly
 - AJAX for the views to show new information
- Interface polishing
 - Better registration process
 - General user interface cleanup

Prospect

- Asynchronous updates
 - Threaded updating of the feeds in the server
 - Controller check for new Tags and Feed-Items regularly
 - AJAX for the views to show new information
- Interface polishing
 - Better registration process
 - General user interface cleanup
- I/O Options
 - OPML import
 - RSS output
 - More input sources

Looking back. . .

What we think we did well. . .

- Steady workflow
- Balance between “Get that feature in” and “Code cleanup”
- Keeping the test-suite ahead of the code
- Keeping the system in working order

... and looking ahead

What we would like to do differently...

- Spend more time on bug-fixing
- Better iteration planning (or at least some planning at all)
- Keeping the test-suite ahead of the code
- Use a better version-control-system (MC2, Git, ...)

On XPForums and SqueakSave

- XPForums
 - User stories were helpful
 - Associations with code could have been better
 - VNC never worked for us
 - Never used code markup, rather discussed that in the meeting

On XPForums and SqueakSave

- XPForums
 - User stories were helpful
 - Associations with code could have been better
 - VNC never worked for us
 - Never used code markup, rather discussed that in the meeting
- SqueakSave
 - A great step up from configuring GLORP
 - Nice concepts (Object»save, Connection pools)
 - Some loose ends and not too helpful error messages
 - ⇒ Database reset was a frequent “Do it”

References

- [1] Yehuda Katz: 6 Steps To Refactoring Rails (for Mere Mortals).

<http://www.engineyard.com/blog/2009/6-steps-to-ref>
(2009)

- [2] Perscheid, M., Tibbe, D., Beck, M., Berger, S., Osburg, P., Eastman, J., Haupt, M., Hirschfeld, R.: An Introduction to Seaside. Software Architecture Group (Hasso-Plattner-Institut) (2008)

- [3] Renggli, L.: Magritte—meta-described web application