

# Technical Debt - Refactor or Rewrite

Tuesday, October 6, 2009

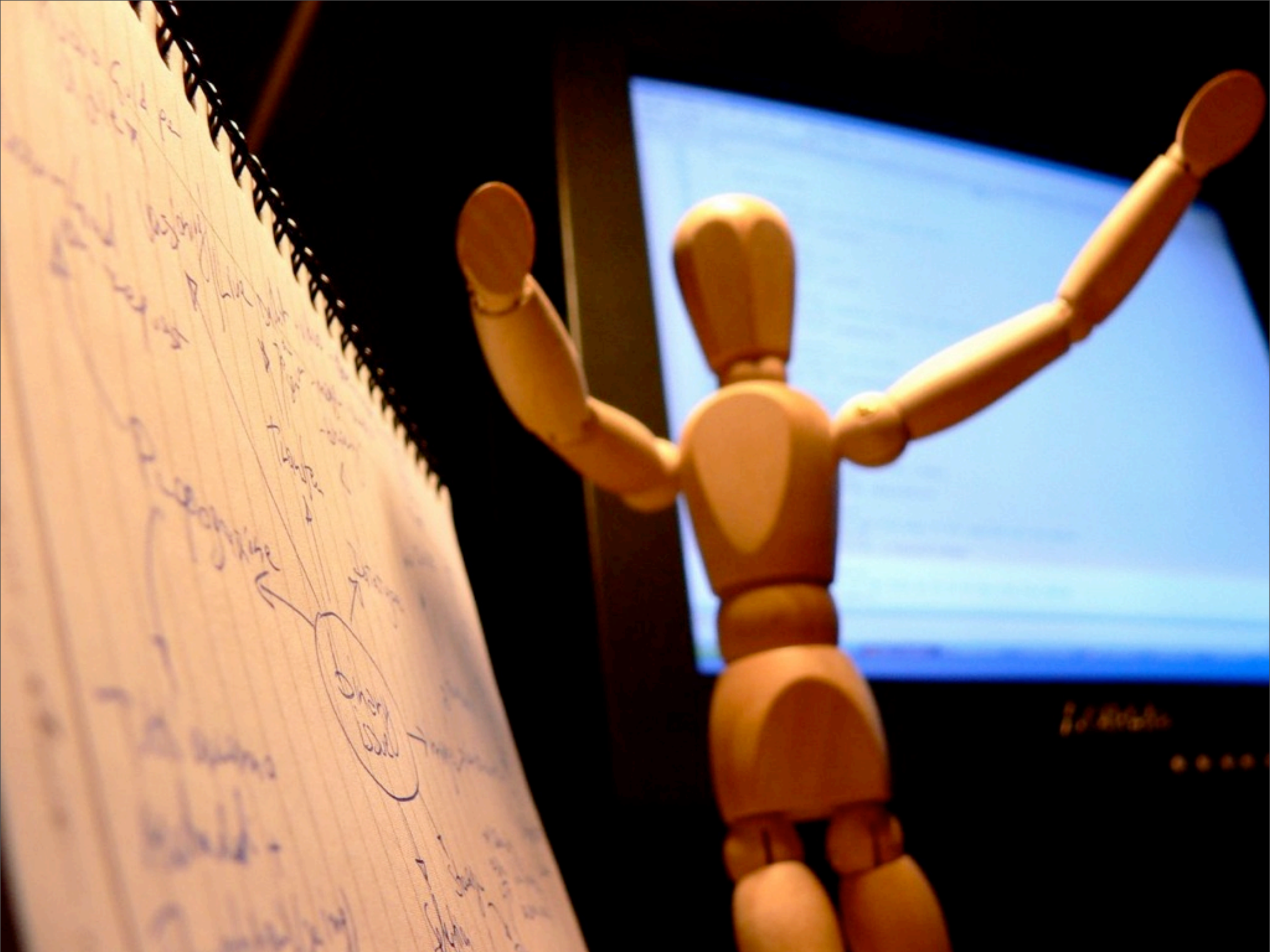
A few disclaimers

YMMV

YAGNI

TATFT

IANAL



Tuesday, October 6, 2009

6

Our job as developers: build stuff to accomplish a goal (make money) as quickly as possible. The as quickly as possible means sometimes we have to make decisions without fully understanding the problem.





Tuesday, October 6, 2009

7

Technical debt is developing software using the knowledge you have now to release sooner. Ward Cunningham coined the term in 1992

“Shipping first time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a rewrite.... The danger occurs when the debt is not repaid. Every minute spent on not-quite-right code counts as [interest](#) on that debt. Entire engineering organizations can be brought to a stand-still under the debt load of an unconsolidated implementation, [object-oriented](#) or otherwise.”





Tuesday, October 6, 2009

8

Paying back the technical debt is done by refactoring when you have a better understanding of what the code should do.





Tuesday, October 6, 2009

9

Technical debt is NOT writing poor code (and poorly tested code).





Tuesday, October 6, 2009

10

Interest is paid until the refactoring is done as it takes longer to complete new features.





Foreclosed

Tuesday, October 6, 2009

11

If you only add new features then you will continue to pay more and more interest. Eventually your debt will outweigh the gains and a rewrite will likely be necessary.

Story: 1999 – Building the first Signature Domains

The desktop app

Adding more services

Abstracting the communications layer



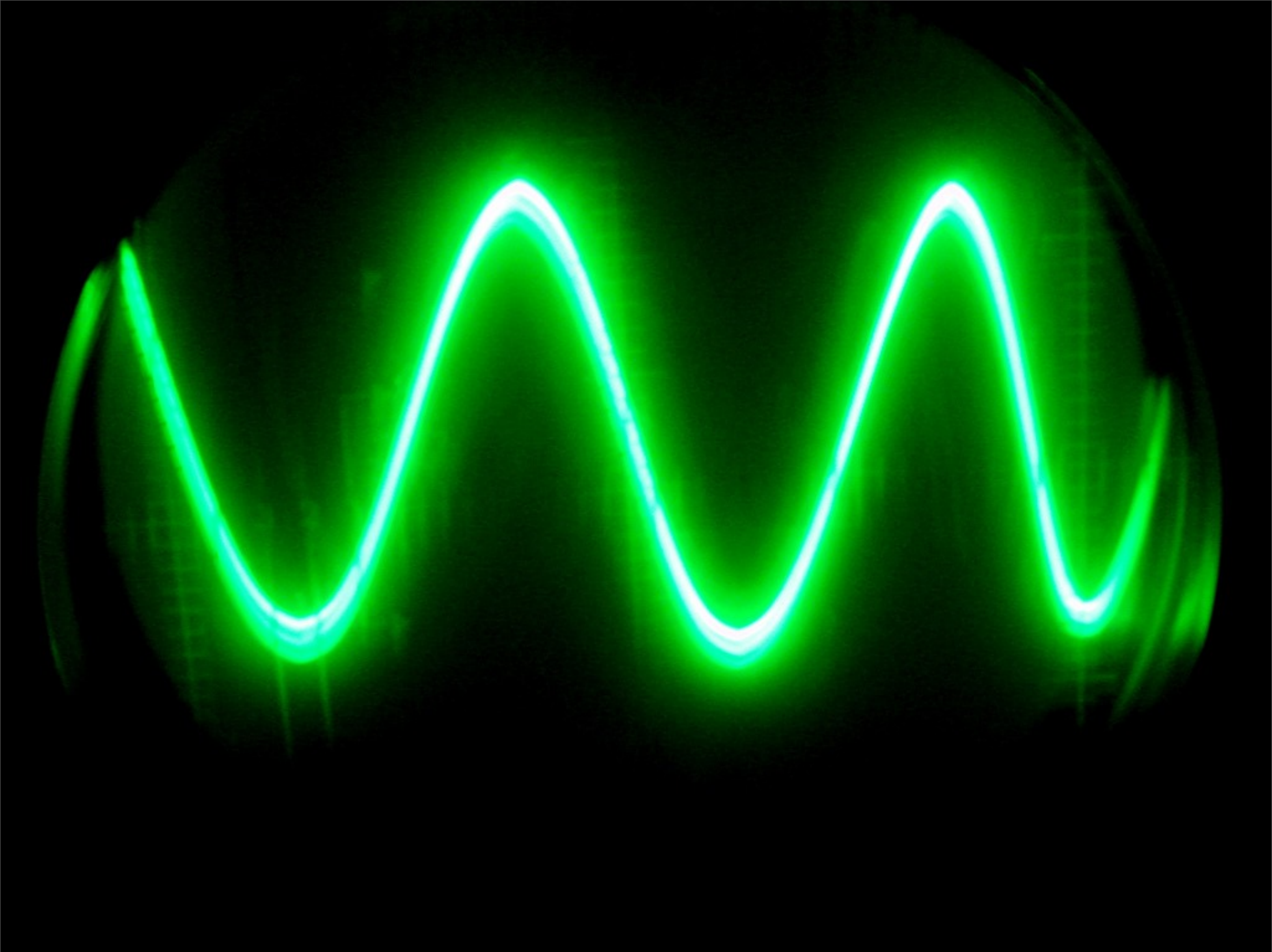


Tuesday, October 6, 2009

12

Techniques for acquiring technical with the intent of repaying it





Tuesday, October 6, 2009

13

Test first, test last, test all the time, just test.



```

# the block.-
def with_queue(name, &block)-
  begin-
    yield queue(name)-
  rescue Exception => e-
    logger.error "An error occurred when communicating with queue #{"
    sleep(30)-
  end-
end-

protected-
# Get the queue provider. Defaults to RightAws::SqsGen2 running
# in multi-thread mode.-
def queue_provider-
  if defined?(Typica)-
    provider ||= Typica::Sqs::QueueService.new(
      credentials["access_key_id"],
      credentials["secret_access_key"]-

```

Tuesday, October 6, 2009

14

Minimize the amount of code written.  
The less code you write the less opportunity for acquiring debt.





Tuesday, October 6, 2009

15

Ensure knowledge is not limited to one person (pair program)  
These are essentially the current best practices for software development





Tuesday, October 6, 2009

Payback time





Tuesday, October 6, 2009

17

Ensure that time is set aside for refactoring





Tuesday, October 6, 2009

18

Remove cruft

Remove commented-out code (Desi)

Remove methods no longer invoked

Remove entire classes and source code files that are not used





Tuesday, October 6, 2009

19

Write tests for remaining code that is untested

Use your client to understand the business logic (Desi)

Writing tests for untested code, when used in conjunction with client knowledge, may result in being able to throw away some code as well





Tuesday, October 6, 2009

20

## Refactor existing code

Replace custom code with built in code (Desi)  
Break up long methods into multiple short methods  
Refactor code for reuse

Story: refactoring a controller





Tuesday, October 6, 2009

21

Sometimes refactoring is not an option. Sometimes a rewrite is the only way to go.





Tuesday, October 6, 2009

22

Key triggers for a rewrite: 0 tests, low-quality code, owner(s) of code are no longer on team





Tuesday, October 6, 2009

23

Warning: you'll fix some problems but create others during a rewrite





Tuesday, October 6, 2009

24

Throw away the first  
If you've already thrown one away then identify pieces to rewrite  
Rewrite with purpose  
Follow the best practices for writing refactorable code

Story: Portable Contacts library





Tuesday, October 6, 2009

25

Summarize:  
Technical debt is inevitable  
Plan to pay it back  
Build to refactor and then refactor  
Rewrite when refactoring is not an option



The perfect software architect - <http://www.flickr.com/photos/ezu/297634534/>  
Dictionary :Time - <http://www.flickr.com/photos/katphotos/2216663973/>  
Vespa Tuning - [http://www.flickr.com/photos/anyone\\_anywhere/149389999/](http://www.flickr.com/photos/anyone_anywhere/149389999/)  
Swamp TV - <http://www.flickr.com/photos/jamesgood/363013819/>  
Loaded - <http://www.flickr.com/photos/fredvdd/352780093/>  
Empty Nest Syndrome - <http://www.flickr.com/photos/bitzcelt/3172219438/>  
Where There's Muck, There's Brass - <http://www.flickr.com/photos/nickwheeleroz/2295584401/>  
Green Sine - <http://www.flickr.com/photos/bdu/381460970/>  
Kevin & Jeff Pair Program - <http://www.flickr.com/photos/igalko/3643220394/>  
It's Payback - <http://www.flickr.com/photos/elpatojo/256573432/>  
Sideways - <http://www.flickr.com/photos/96dpi/2202694688/>  
Cooking Knives - <http://www.flickr.com/photos/nickwheeleroz/904301529/>  
Silken Circles - <http://www.flickr.com/photos/piper/240951732/>  
Breaking the Ice - <http://www.flickr.com/photos/marcelgermain/2078076913/>  
Put on a Happy Face - <http://www.flickr.com/photos/44442915@N00/3981675322/>  
Nude Sunbathing - <http://www.flickr.com/photos/tochis/530311934/>  
Sometimes Them's The Breaks - <http://www.flickr.com/photos/78215847@N00/352958077/>  
McDonald's Hawaiian Trash Cans - <http://www.flickr.com/photos/phillipstewart/3767177311/>  
In Search of Lost Time - <http://www.flickr.com/photos/bogenfreund/556656621/>



Anthony Eden  
<http://anthony.mp>

12 step program

As summarized by the American Psychological Association, the process involves the following:

- admitting that one cannot control one's addiction or compulsion;
- recognizing a greater power that can give strength; (ruby)
- examining past errors with the help of a sponsor (experienced member); (senior developer)
- making amends for these errors; (refactor or rewrite)
- learning to live a new life with a new code of behavior; (best practices)
- helping others that suffer from the same addictions or compulsions. (this talk)