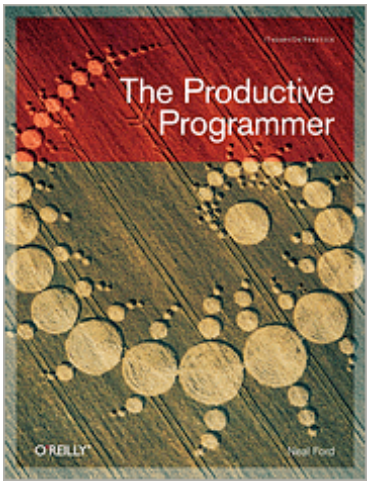
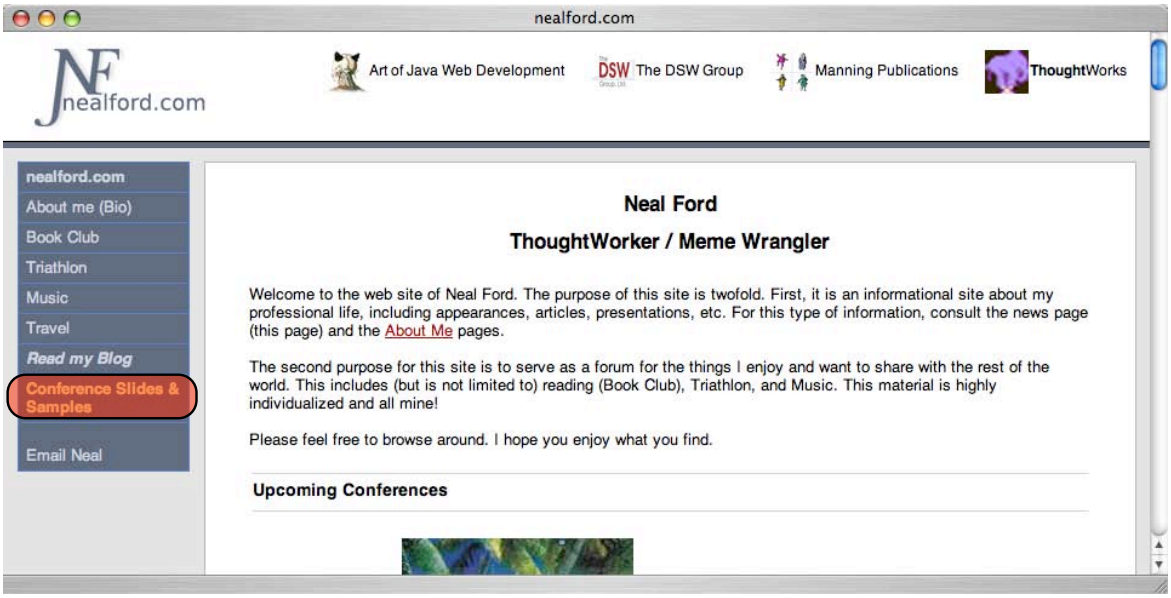
the productive programmer: mechanics



NEAL FORD software architect / meme wrangler

ThoughtWorks

nford@thoughtworks.com
3003 Summit Boulevard, Atlanta, GA 30319
www.nealford.com
www.thoughtworks.com
memeagora.blogspot.com

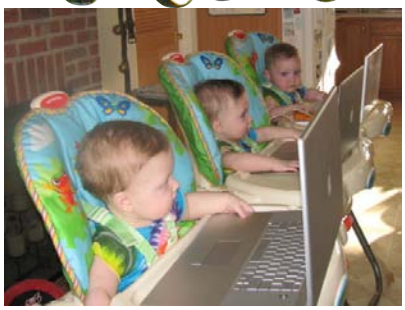


where did this topic come from?

?

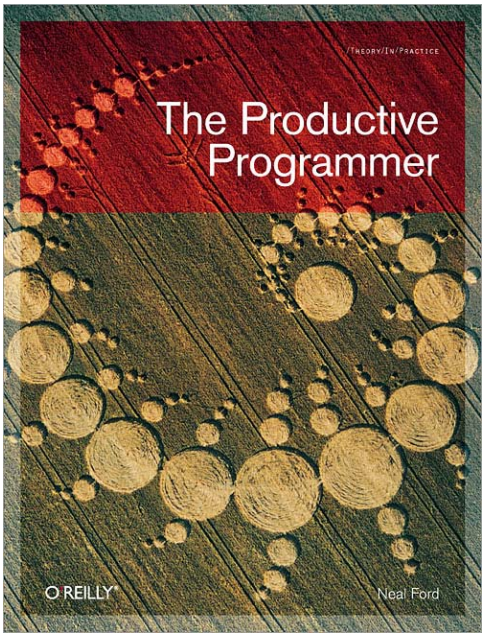
where did this topic come from?

```
[nealford] ~ ]=> ls
2print                                  Video
Applications                            archive_of_docs.zip
Archive                                 bin
Desktop                                 crypt.sparseimage
Documents                               derby.log
Downloads                               dev
Images                                  docbook
Library                                 docs
Mingle                                  en_windows_vista_x86_dvd_X12-34293.iso
Movies                                  nfjs
Music                                   print
Neal_Ford-LOP_Shifting_Paradigms-slides.pdf reference
NetBeansProjects                        software installs.sparseimage
Notes702Mac.dmg                         talks
PDFs                                    temp
PartitionMagic805AllWin_English.ZIP     virtual machines
Pictures                                vista keys.txt
Public                                  web
Sites                                   work
[nealford] ~ ]=> cd ~/work/nealford/
[nealford] ~/work/nealford ]=>  svn st | grep '^\?' | tr '^\?' ' ' | sed 's/[ ]*//' | se
d 's/[ ]/\\ /g' | xargs svn add
```



CodeSherpas

where did this topic come from?



what i cover:

acceleration

- doing stuff faster

focus

- killing distractions

automation

- getting your computer to work harder

canonicality

- applying the **dry** principle

acceleration

typing is faster than navigation

o/s accelerators

windows explorer address bar (**`alt-d`**)

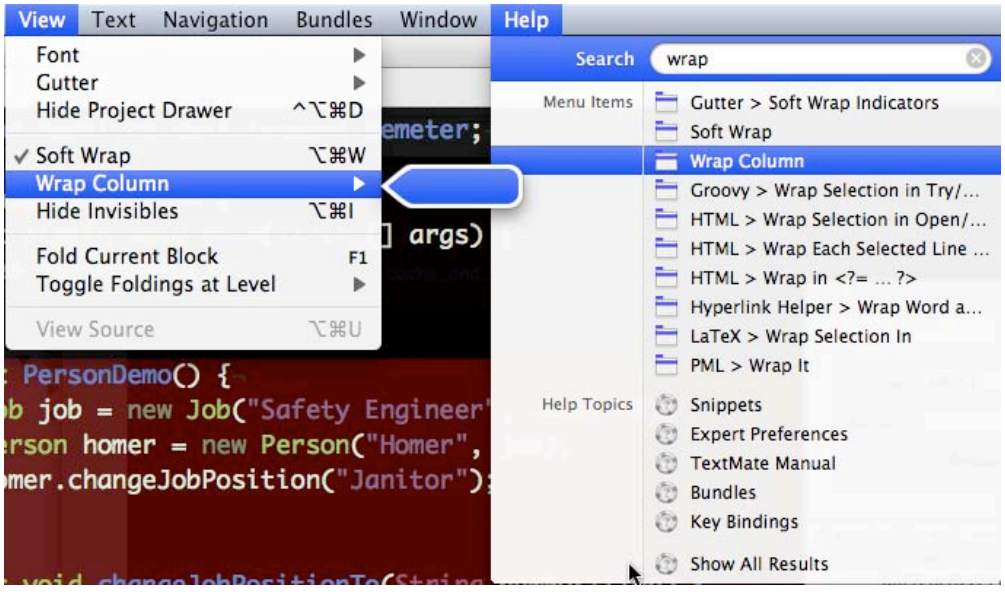
finder (**`apple-shift-g`**)

firefox

/ searching

number-fox plugin

leopard smart help



clipboards

why do operating systems have only 1 clipboard with 1 entry????

clcl 

iClip ($$)

jump cut

mac os x : jumpcut

context switching eats time

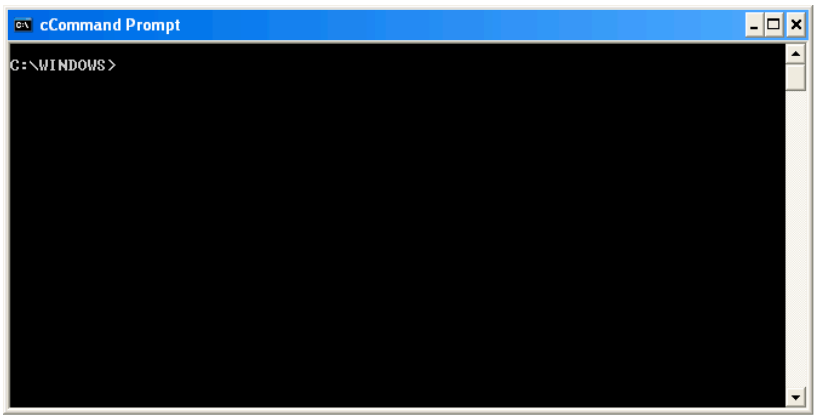
there and back

pushd pushes a directory on the stack

popd pops it back off

```
C:\temp>pushd \MyDocuments\Documents\dev\haskell

C:\MyDocuments\Documents\dev\haskell>dir
02/13/2006  12:12 AM    <DIR>          .
02/13/2006  12:12 AM    <DIR>          ..
02/13/2006  12:12 AM    <DIR>          nfjs_functionallangs_haskell
               0 File(s)              0 bytes
               3 Dir(s)  11,743,178,752 bytes free

C:\MyDocuments\Documents\dev\haskell>popd
```

pushd/popd



command prompts

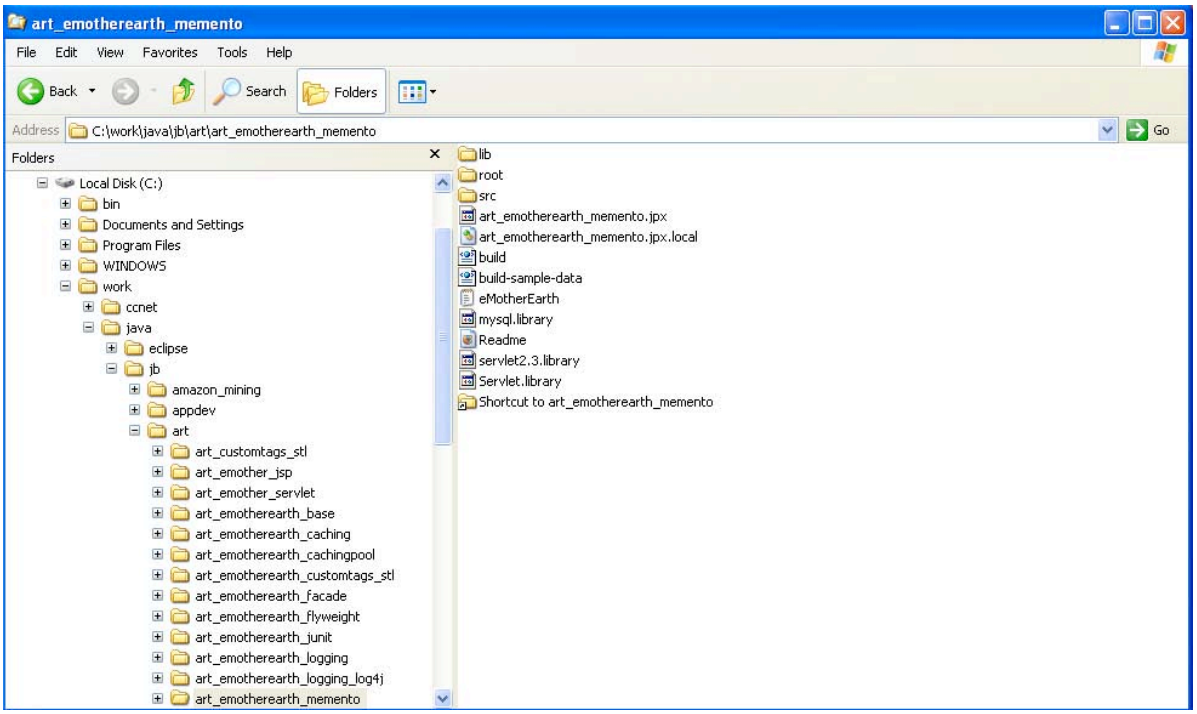
graphical explorers better for some things....

...command line better for others

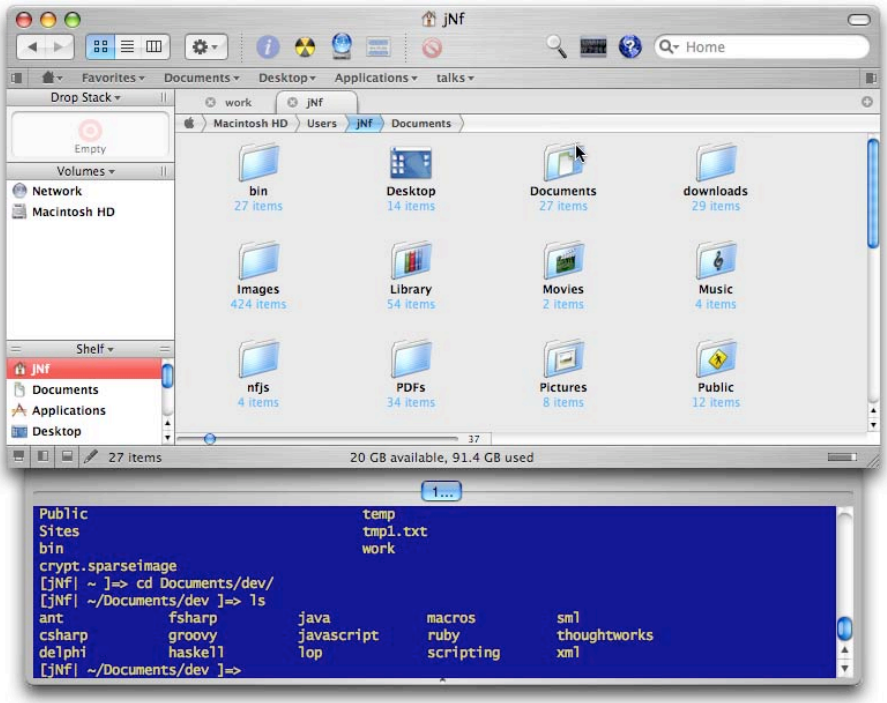
command prompt here power toy

bash here (cygwin)

cmd prompt explorer bar



path finder



how many of you have written an application for heads-down data entry personnel?

when coding, always prefer keyboard to mouse

learning shortcuts

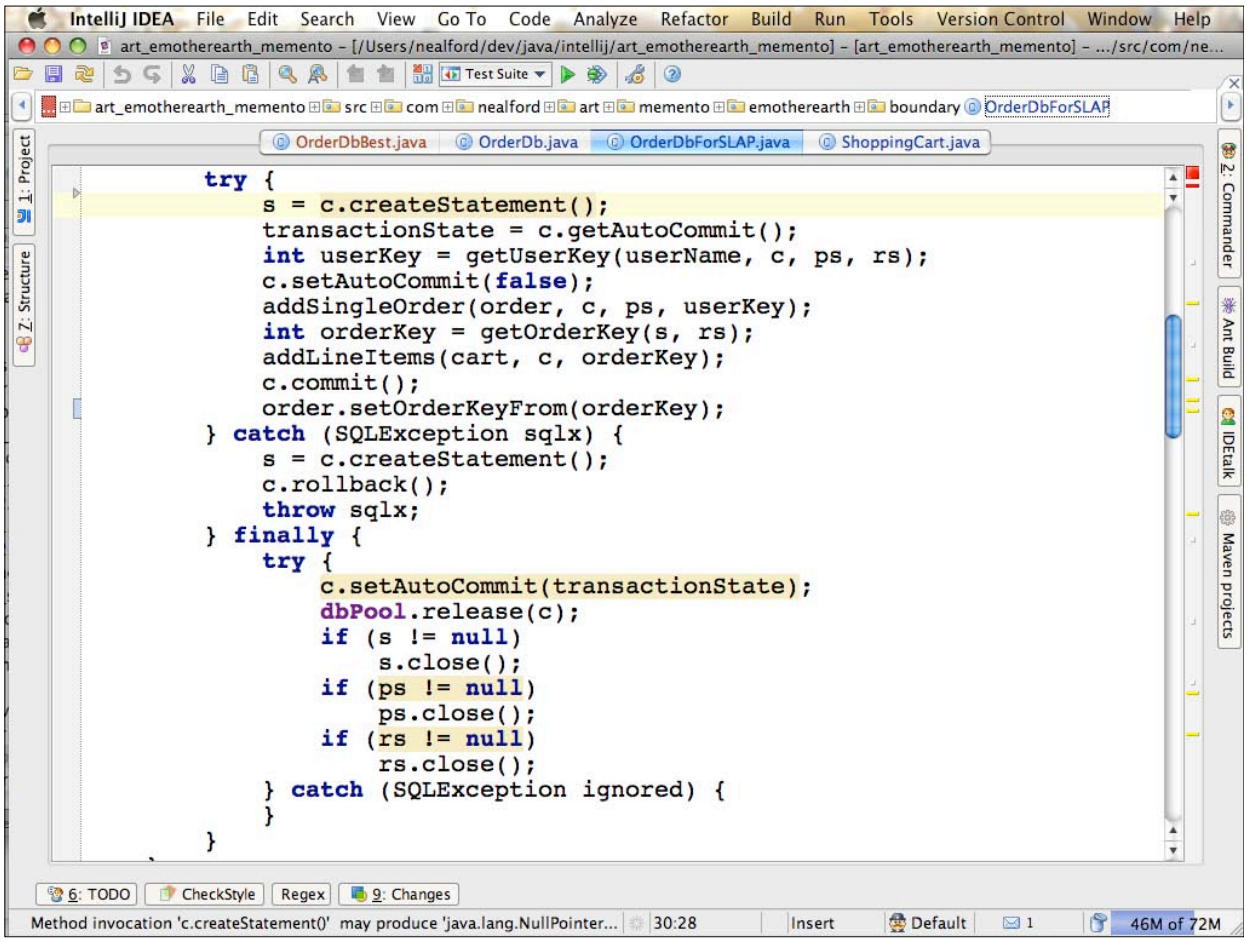
make yourself use the shortcut even if you've gotten there another way

have someone/something pester you about it

pair programmer

key promoter plug-in for intellij

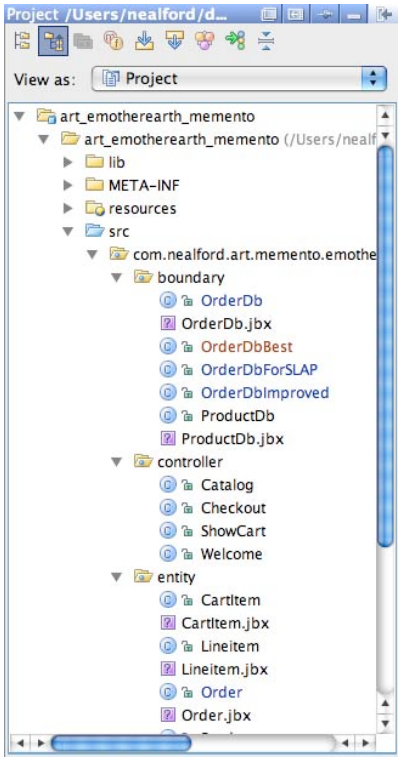
mousefeed for eclipse



learning shortcuts

repeat them to yourself

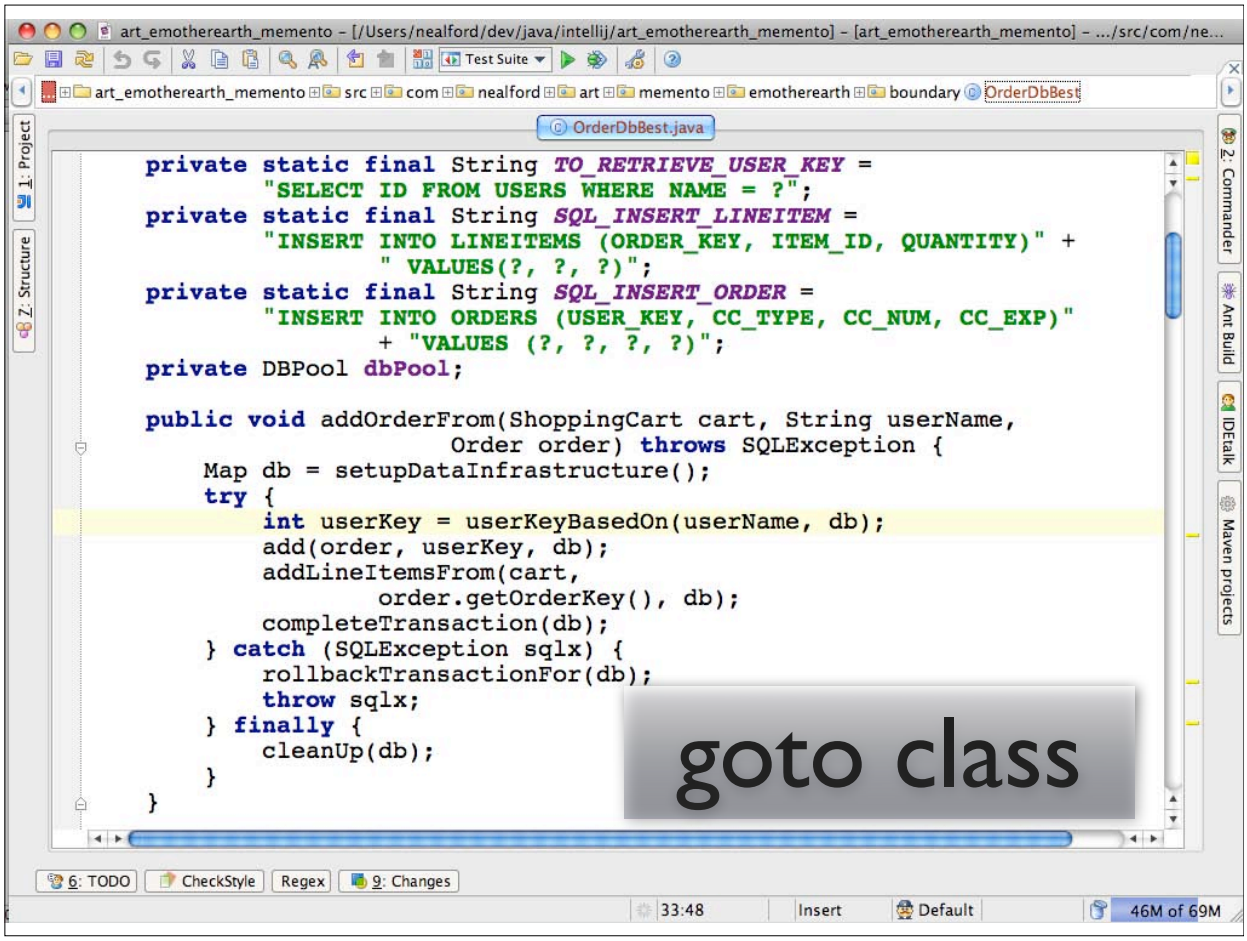
flash cards

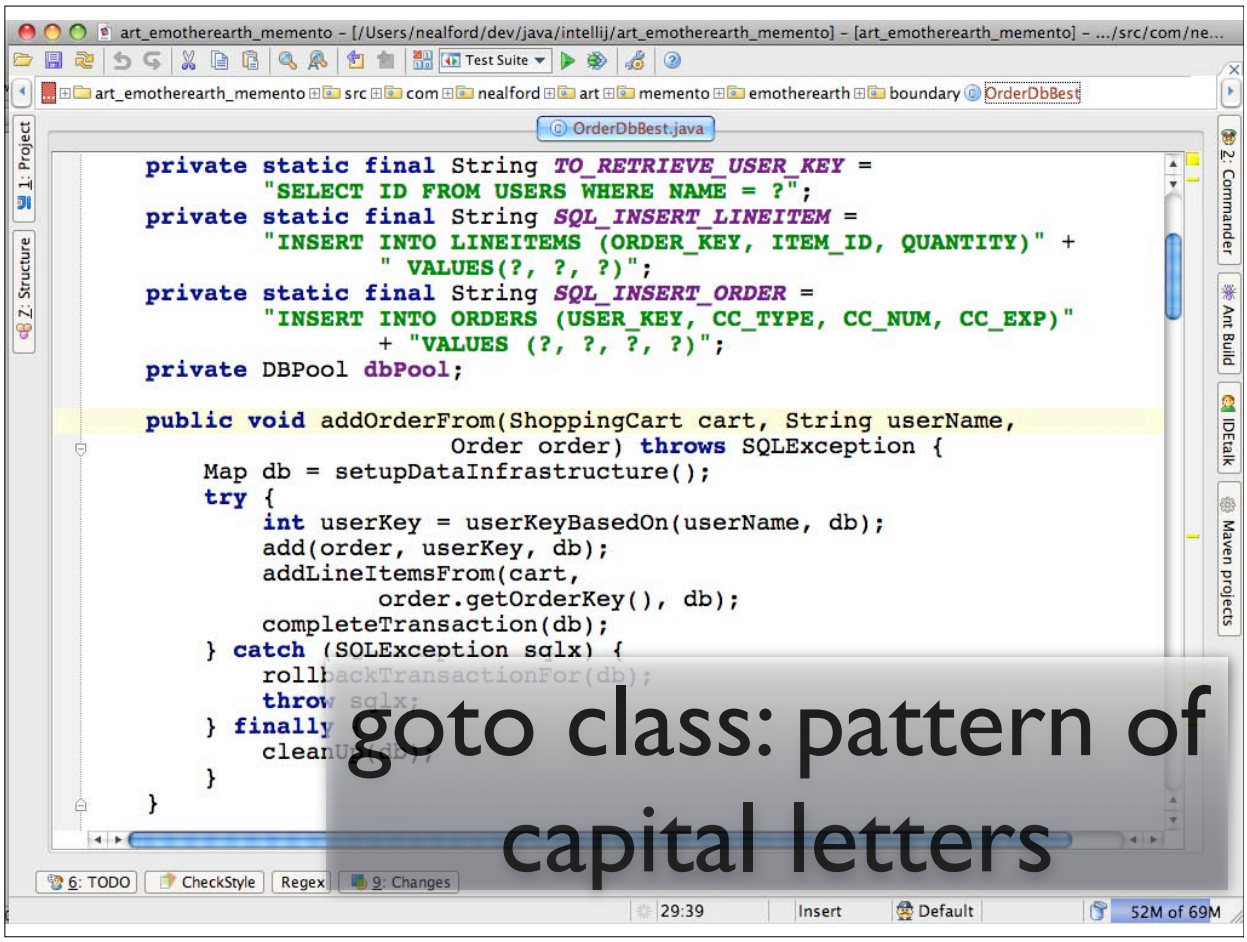


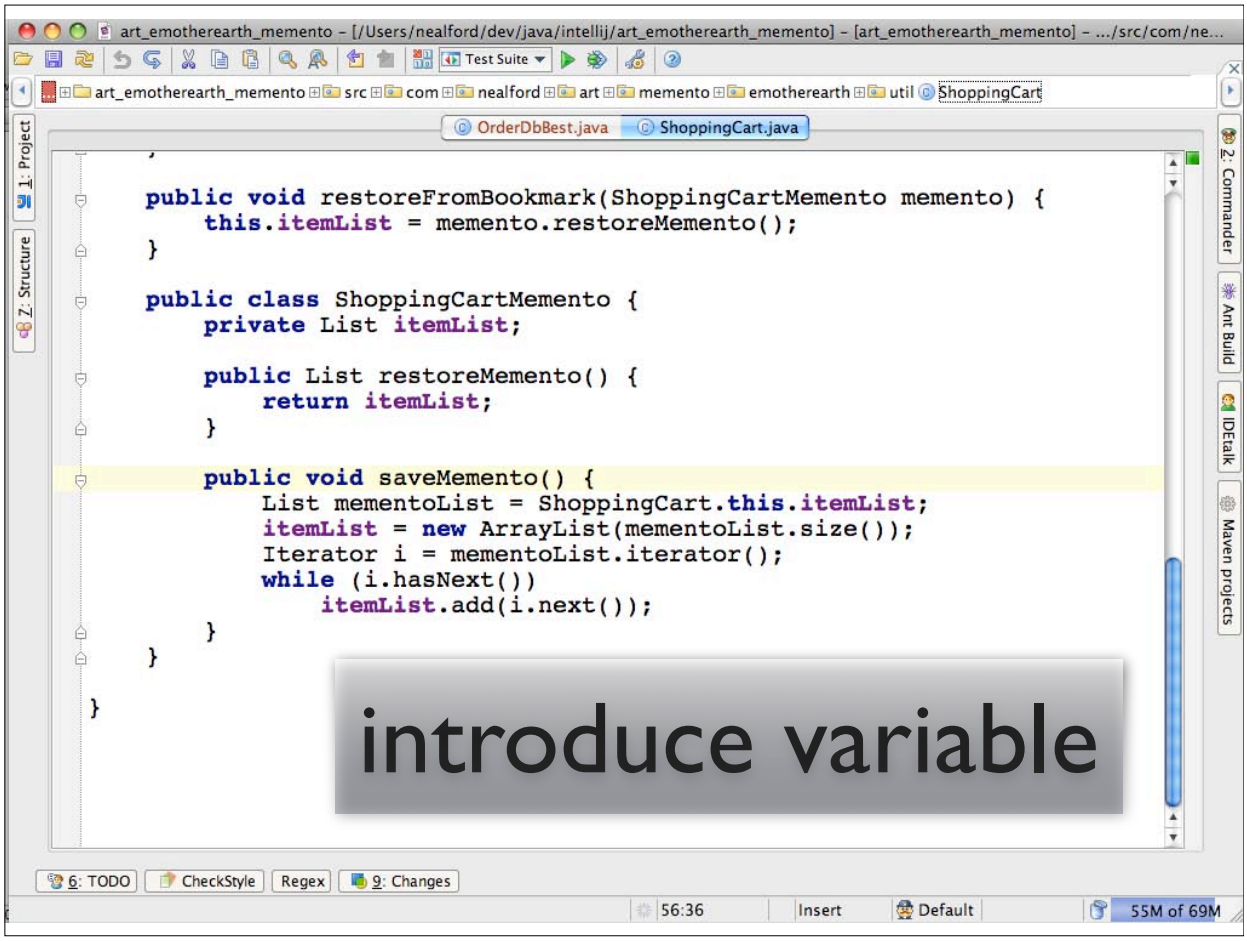
all our hierarchies are too deep:

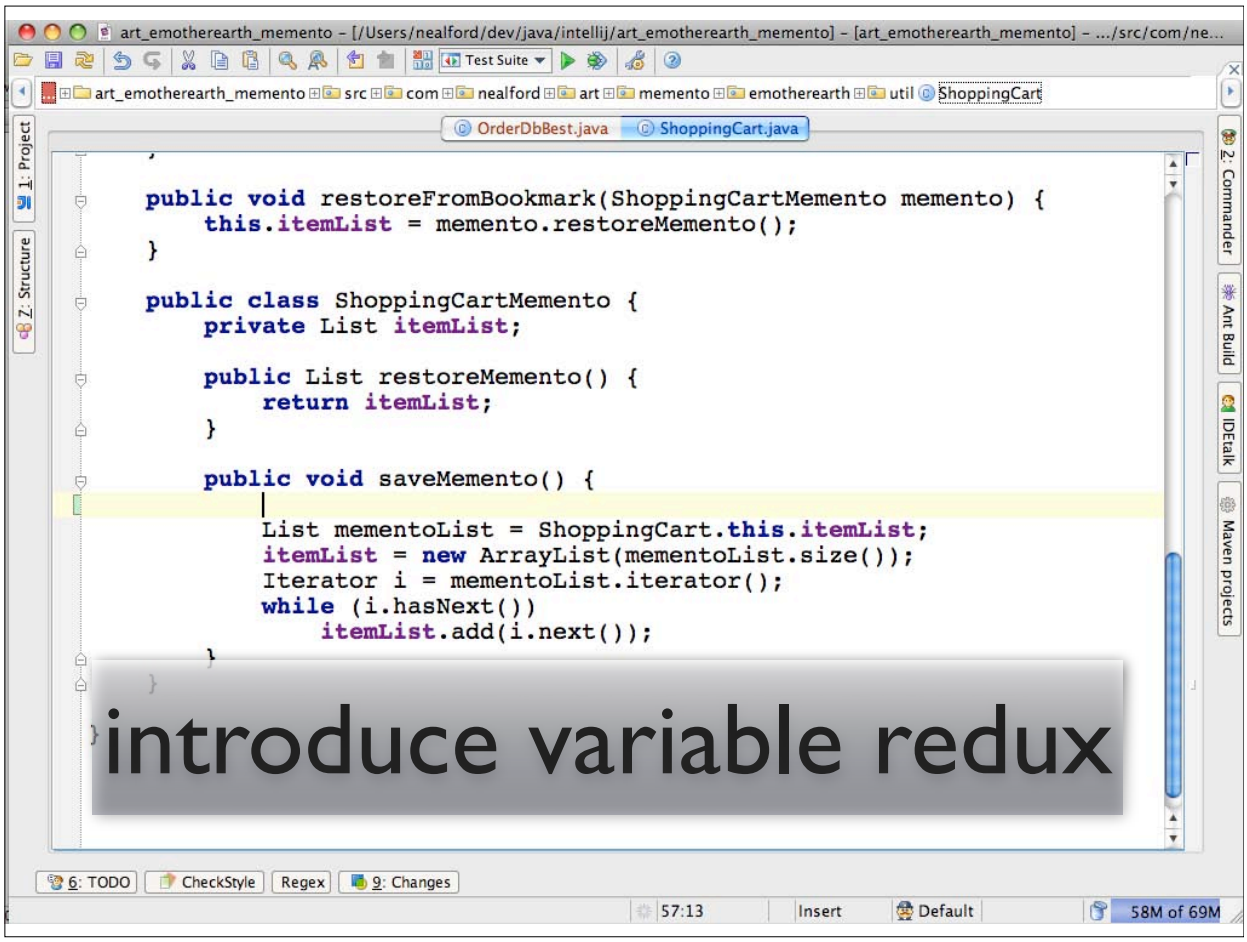
file system

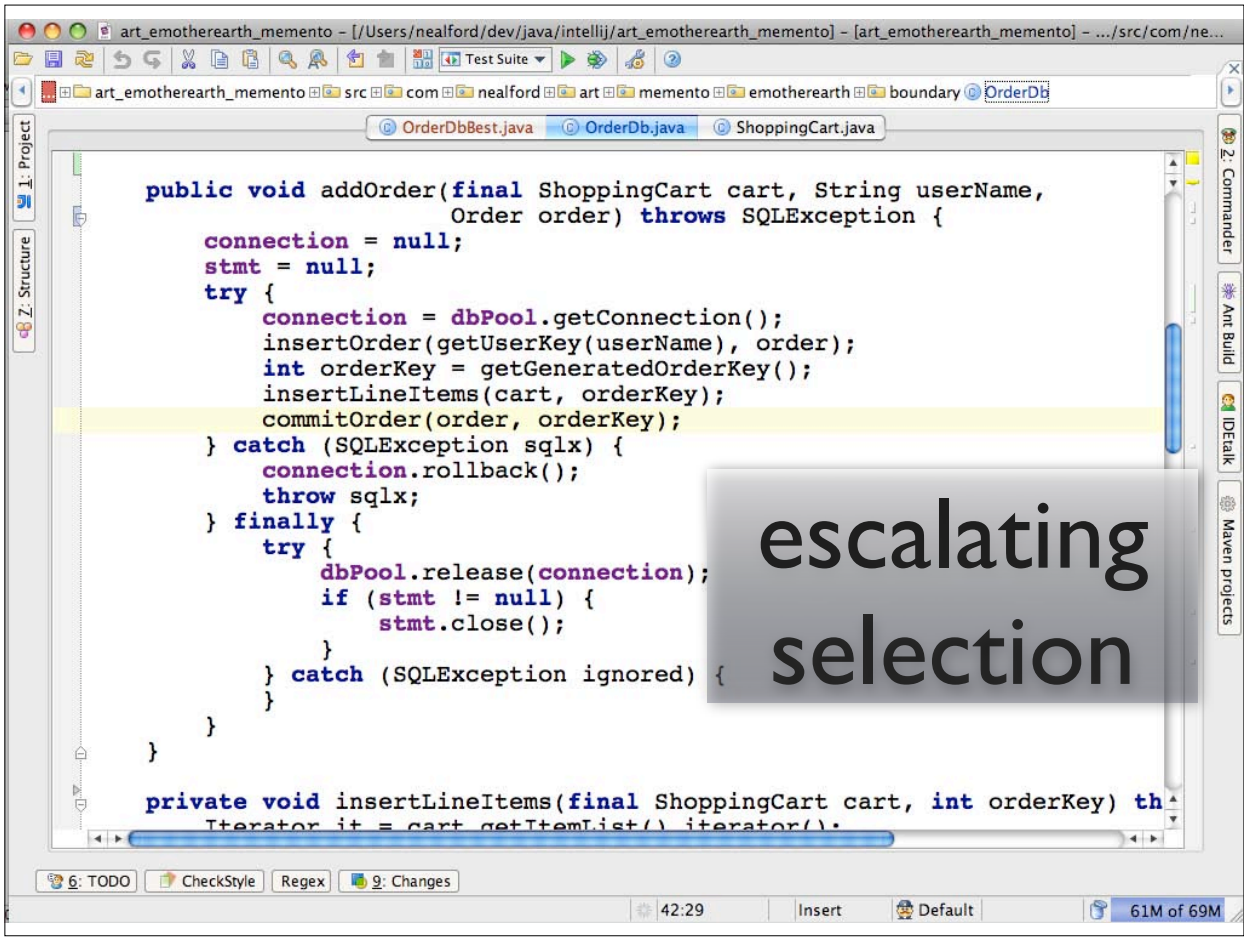
packages

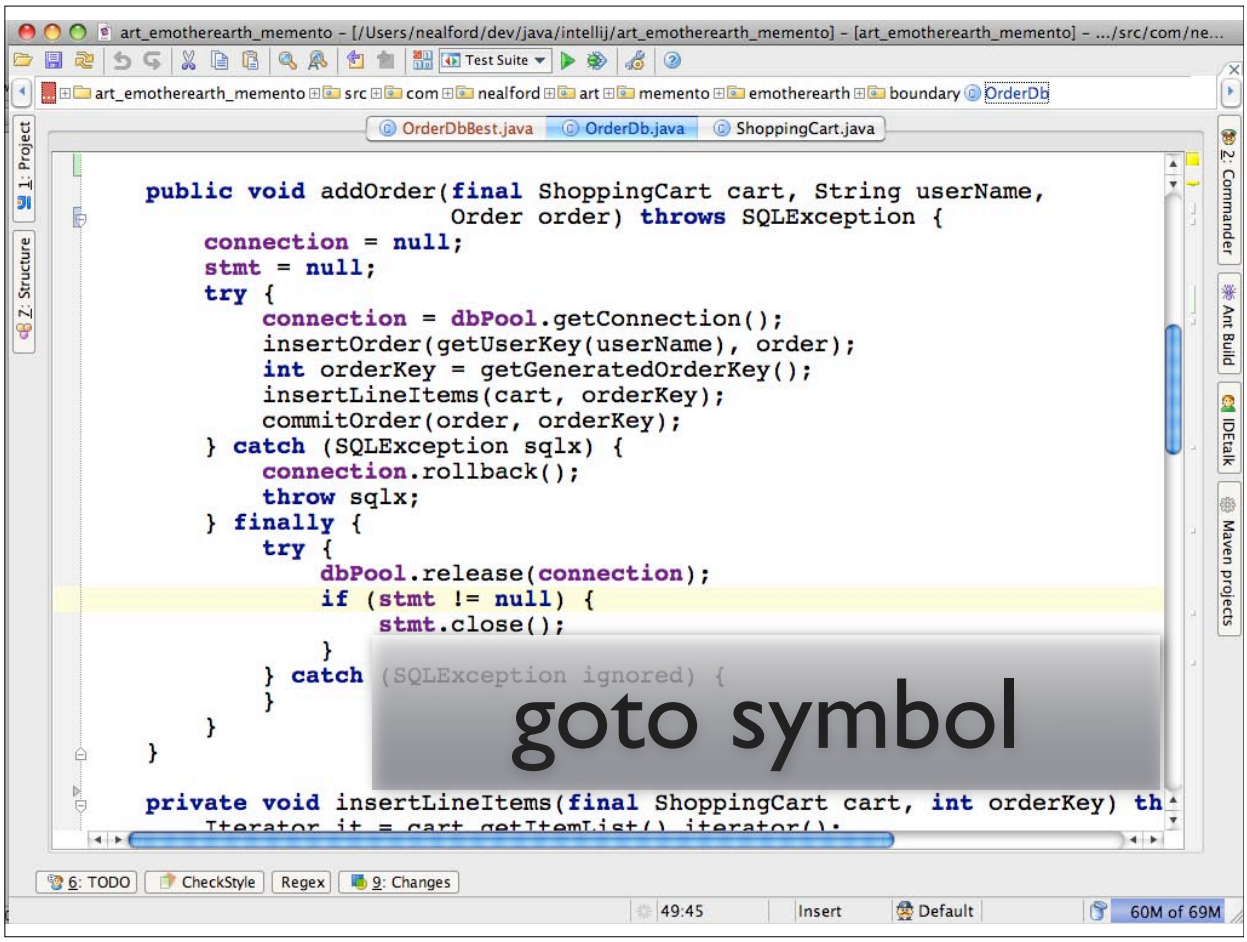

art_emotherearth_memento – [/Users/nealford/dev/java/intellij/art_emotherearth_memento] – [art_emotherearth_memento] – .../src/com/ne...
Test Suite
art_emotherearth_memento src com nealford art memento emotherearth boundary OrderDbBest
OrderDbBest.java
private static final String TO_RETRIEVE_USER_KEY =
"SELECT ID FROM USERS WHERE NAME = ?";
private static final String SQL_INSERT_LINEITEM =
"INSERT INTO LINEITEMS (ORDER_KEY, ITEM_ID, QUANTITY)" +
" VALUES(?, ?, ?)";
private static final String SQL_INSERT_ORDER =
"INSERT INTO ORDERS (USER_KEY, CC_TYPE, CC_NUM, CC_EXP)"
+ "VALUES (?, ?, ?, ?)";
private DBPool dbPool;
public void addOrderFrom(ShoppingCart cart, String userName,
Order order) throws SQLException {
Map db = setupDataInfrastructure();
try {
int userKey = userKeyBasedOn(userName, db);
add(order, userKey, db);
addLineItemsFrom(cart,
order.getOrderKey(), db);
completeTransaction(db);
} catch (SQLException sqlx) {
rollbackTransactionFor(db);
throw sqlx;
} finally {
cleanUp(db);
}
}
goto class
6: TODO
CheckStyle
Regex
9: Changes
33:48
Insert
Default
46M of 69M


art_emotherearth_memento – [/Users/nealford/dev/java/intellij/art_emotherearth_memento] – [art_emotherearth_memento] – .../src/com/ne...
Test Suite
art_emotherearth_memento src com nealford art memento emotherearth boundary OrderDbBest
OrderDbBest.java
private static final String TO_RETRIEVE_USER_KEY =
"SELECT ID FROM USERS WHERE NAME = ?";
private static final String SQL_INSERT_LINEITEM =
"INSERT INTO LINEITEMS (ORDER_KEY, ITEM_ID, QUANTITY)" +
" VALUES(?, ?, ?)";
private static final String SQL_INSERT_ORDER =
"INSERT INTO ORDERS (USER_KEY, CC_TYPE, CC_NUM, CC_EXP)"
+ "VALUES (?, ?, ?, ?)";
private DBPool dbPool;
public void addOrderFrom(ShoppingCart cart, String userName,
Order order) throws SQLException {
Map db = setupDataInfrastructure();
try {
int userKey = userKeyBasedOn(userName, db);
add(order, userKey, db);
addLineItemsFrom(cart,
order.getOrderKey(), db);
completeTransaction(db);
goto class: pattern of capital letters
6: TODO
CheckStyle
Regex
9: Changes
29:39
Insert
Default
52M of 69M


art_emotherearth_memento – [/Users/nealford/dev/java/intellij/art_emotherearth_memento] – [art_emotherearth_memento] – .../src/com/ne...
Test Suite
art_emotherearth_memento src com nealford art memento emotherearth util ShoppingCart
OrderDbBest.java
ShoppingCart.java
public void restoreFromBookmark(ShoppingCartMemento memento) {
this.itemList = memento.restoreMemento();
}
public class ShoppingCartMemento {
private List itemList;
public List restoreMemento() {
return itemList;
}
public void saveMemento() {
List mementoList = ShoppingCart.this.itemList;
itemList = new ArrayList(mementoList.size());
Iterator i = mementoList.iterator();
while (i.hasNext())
itemList.add(i.next());
}
}
}
introduce variable
1: Project
Z: Structure
2: Commander
Ant Build
IDEtalk
Maven projects
6: TODO
CheckStyle
Regex
9: Changes
56:36
Insert
Default
55M of 69M


art_emotherearth_memento – [/Users/nealford/dev/java/intellij/art_emotherearth_memento] – [art_emotherearth_memento] – .../src/com/ne...
Test Suite
art_emotherearth_memento src com nealford art memento emotherearth util ShoppingCart
OrderDbBest.java
ShoppingCart.java
public void restoreFromBookmark(ShoppingCartMemento memento) {
this.itemList = memento.restoreMemento();
}
public class ShoppingCartMemento {
private List itemList;
public List restoreMemento() {
return itemList;
}
public void saveMemento() {
List mementoList = ShoppingCart.this.itemList;
itemList = new ArrayList(mementoList.size());
Iterator i = mementoList.iterator();
while (i.hasNext())
itemList.add(i.next());
}
}
introduce variable redux
1: Project
Z: Structure
2: Commander
Ant Build
IDEtalk
Maven projects
6: TODO
CheckStyle
Regex
9: Changes
57:13
Insert
Default
58M of 69M


art_emotherearth_memento – [/Users/nealford/dev/java/intellij/art_emotherearth_memento] – [art_emotherearth_memento] – .../src/com/ne...
Test Suite
art_emotherearth_memento src com nealford art memento emotherearth boundary OrderDb
OrderDbBest.java
OrderDb.java
ShoppingCart.java
1: Project
Z: Structure
2: Commander
Ant Build
IDEtalk
Maven projects
public void addOrder(final ShoppingCart cart, String userName,
Order order) throws SQLException {
connection = null;
stmt = null;
try {
connection = dbPool.getConnection();
insertOrder(getUserKey(userName), order);
int orderKey = getGeneratedOrderKey();
insertLineItems(cart, orderKey);
commitOrder(order, orderKey);
} catch (SQLException sqlx) {
connection.rollback();
throw sqlx;
} finally {
try {
dbPool.release(connection);
if (stmt != null) {
stmt.close();
}
} catch (SQLException ignored) {
}
}
}
private void insertLineItems(final ShoppingCart cart, int orderKey) th
escalating selection
6: TODO
CheckStyle
Regex
9: Changes
42:29
Insert
Default
61M of 69M


art_emotherearth_memento – [/Users/nealford/dev/java/intellij/art_emotherearth_memento] – [art_emotherearth_memento] – .../src/com/ne...
Test Suite
art_emotherearth_memento src com nealford art memento emotherearth boundary OrderDb
OrderDbBest.java
OrderDb.java
ShoppingCart.java
1: Project
Z: Structure
2: Commander
Ant Build
IDEtalk
Maven projects
public void addOrder(final ShoppingCart cart, String userName,
Order order) throws SQLException {
connection = null;
stmt = null;
try {
connection = dbPool.getConnection();
insertOrder(getUserKey(userName), order);
int orderKey = getGeneratedOrderKey();
insertLineItems(cart, orderKey);
commitOrder(order, orderKey);
} catch (SQLException sqlx) {
connection.rollback();
throw sqlx;
} finally {
try {
dbPool.release(connection);
if (stmt != null) {
stmt.close();
}
} catch (SQLException ignored) {
}
}
}
private void insertLineItems(final ShoppingCart cart, int orderKey) th
goto symbol
6: TODO
CheckStyle
Regex
9: Changes
49:45
Insert
Default
60M of 69M

some choice shortcuts

	intellij	***eclipse***
goto class	ctrl-n	ctrl-shift-t
introduce variable	ctrl-alt-v	alt-shift-l
escalating selection	ctrl-w	alt-shift-up
recently edited files	ctrl-e	n/a (ctrl-e)
symbol list	alt-ctrl-shift-n	ctrl-o
incremental search	alt-f3	ctrl-j

live templates

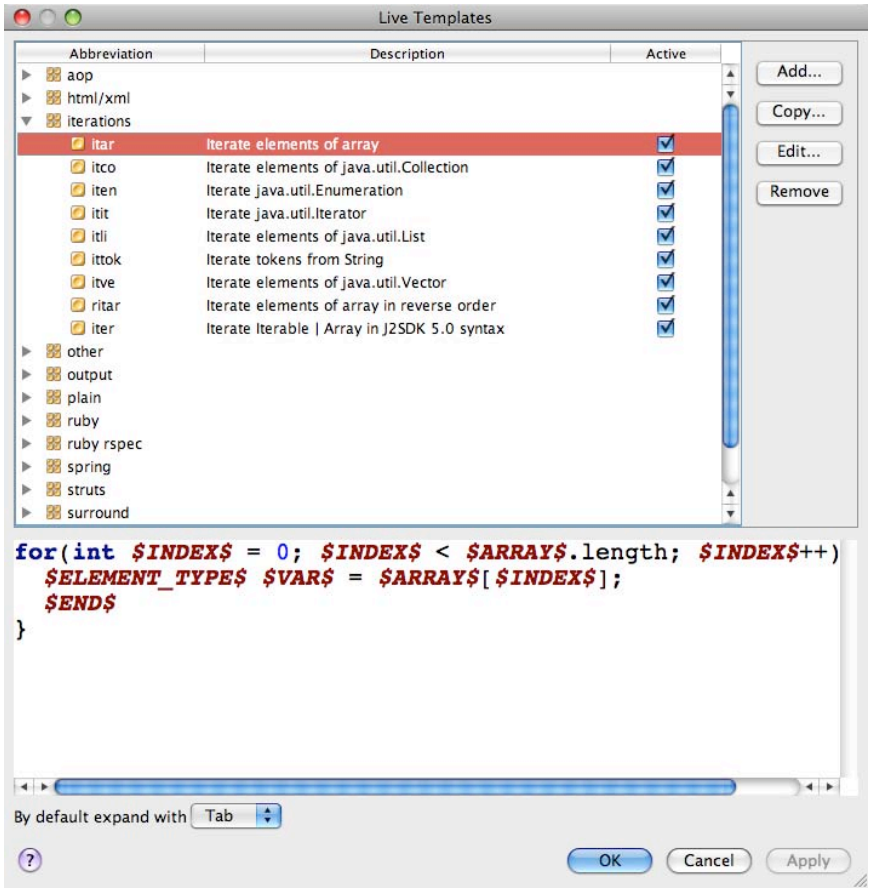
all major ide's and coding text editors

parameter substitution, default values, repeating values

learn the language of your template engine

- velocity in intellij
- bash for textmate/e editor

intellij templates



every time you type
something for the 3rd time,
templatize it

key macro tools

live templates at the o/s level

auto-hot key 

textexpander

typinator 

textexpander

don’t type the same
commands over and over

simple stuff

get a comfortable chair!

dual monitors...

...sitting immediately in front of you

administrator privilege for the o/s

good keyboard

insidious distractions

modern office environments are terrible for knowledge workers

too much out of context noise

how many people here work in cube land?

war rooms

locus of attention

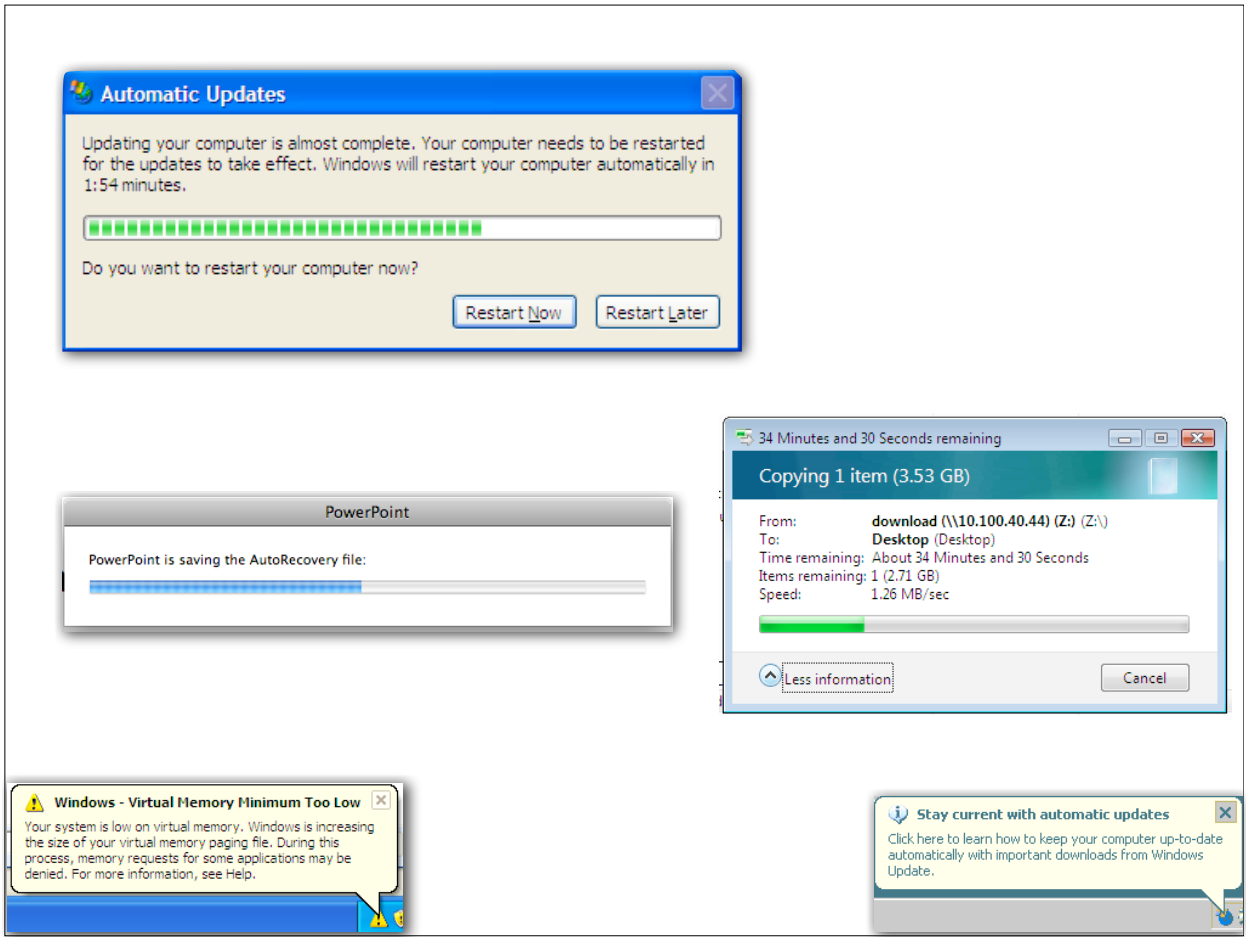
in *the humane interface*, jef raskin describes *locus of attention*

anything that happens outside your locus of attention breaks *flow*

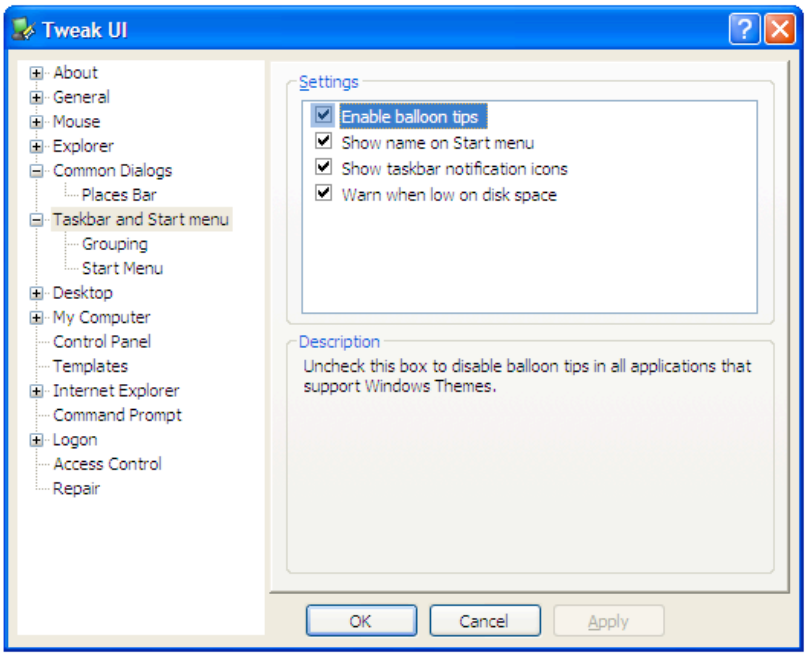
in *flow*, michael csikszentmihalyi describes flow state

total concentration

time disappears



killing balloon tips



screen dimmers

automatically makes your background dark after a set time

jedi concentrate 

doodim 

the higher the level of concentration, the denser the ideas

the easy stuff

turn off notifications

don't keep email open

turn off instant messaging

put on headphones

create office "quiet time"

focus techniques

search > navigation

all developer hierarchies are too deep

- file system
- package/namespace
- documentation

what worked well with 20 mb hard drives fails with 200 gb

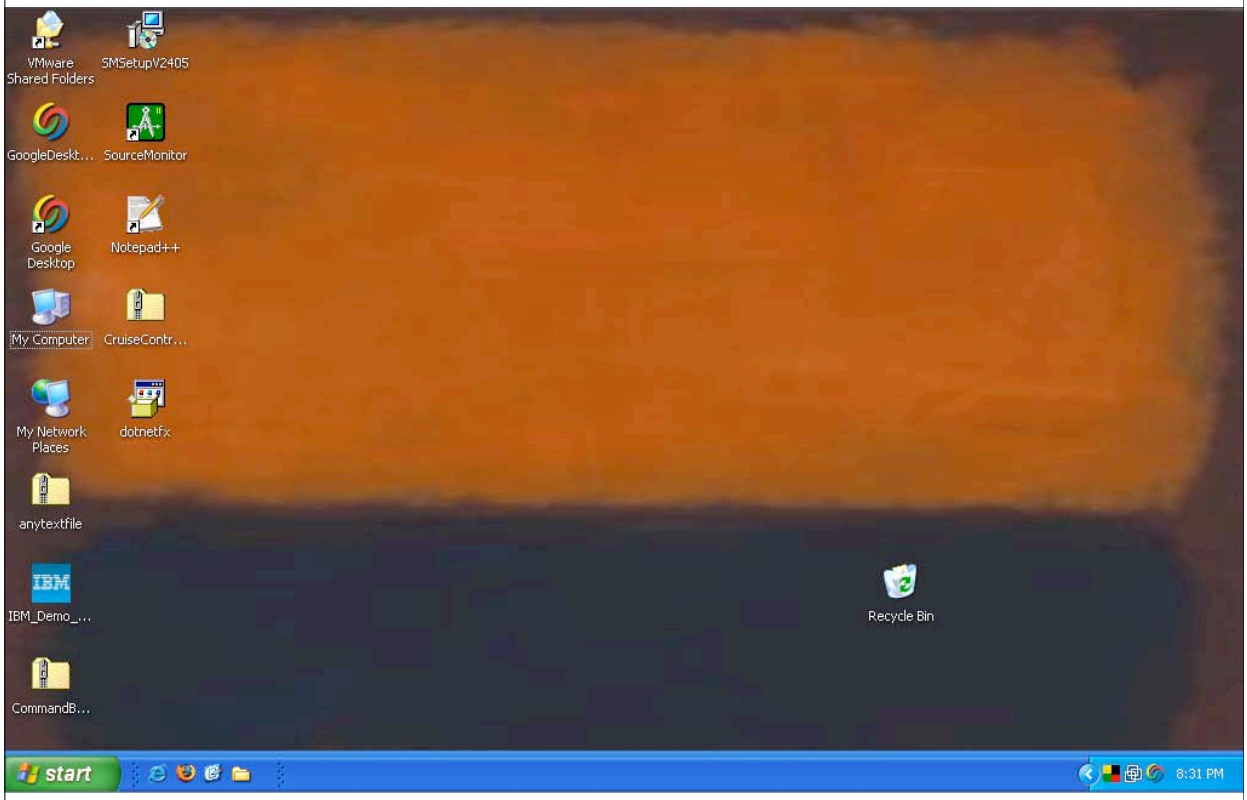
desktop search

built into modern operating systems

retro-fittable in older ones

google desktop search

larry's "any text file" indexer



replace file hierarchy
navigation with search

rooted views

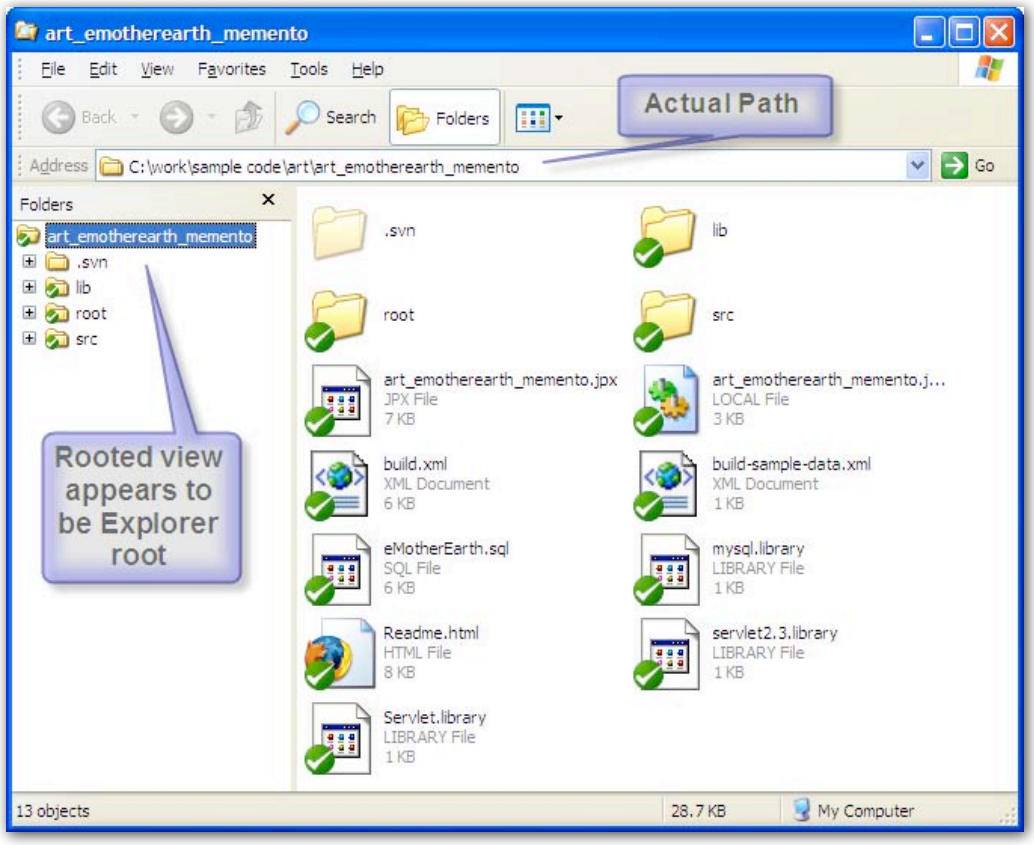
specialized explorer view

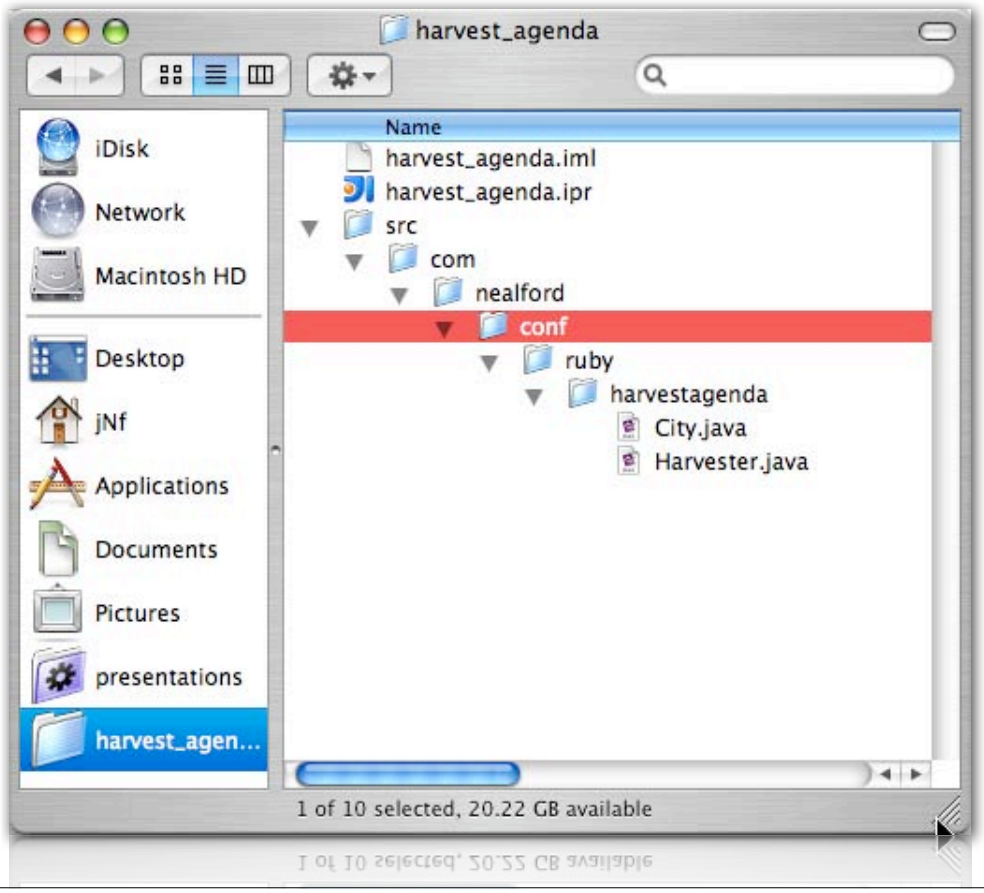
especially good for directory-based version control

rooted view == project explorer

create a shortcut:

```
C:\WINDOWS\explorer.exe /e,/root,c:\work\project
```





use virtual desktops

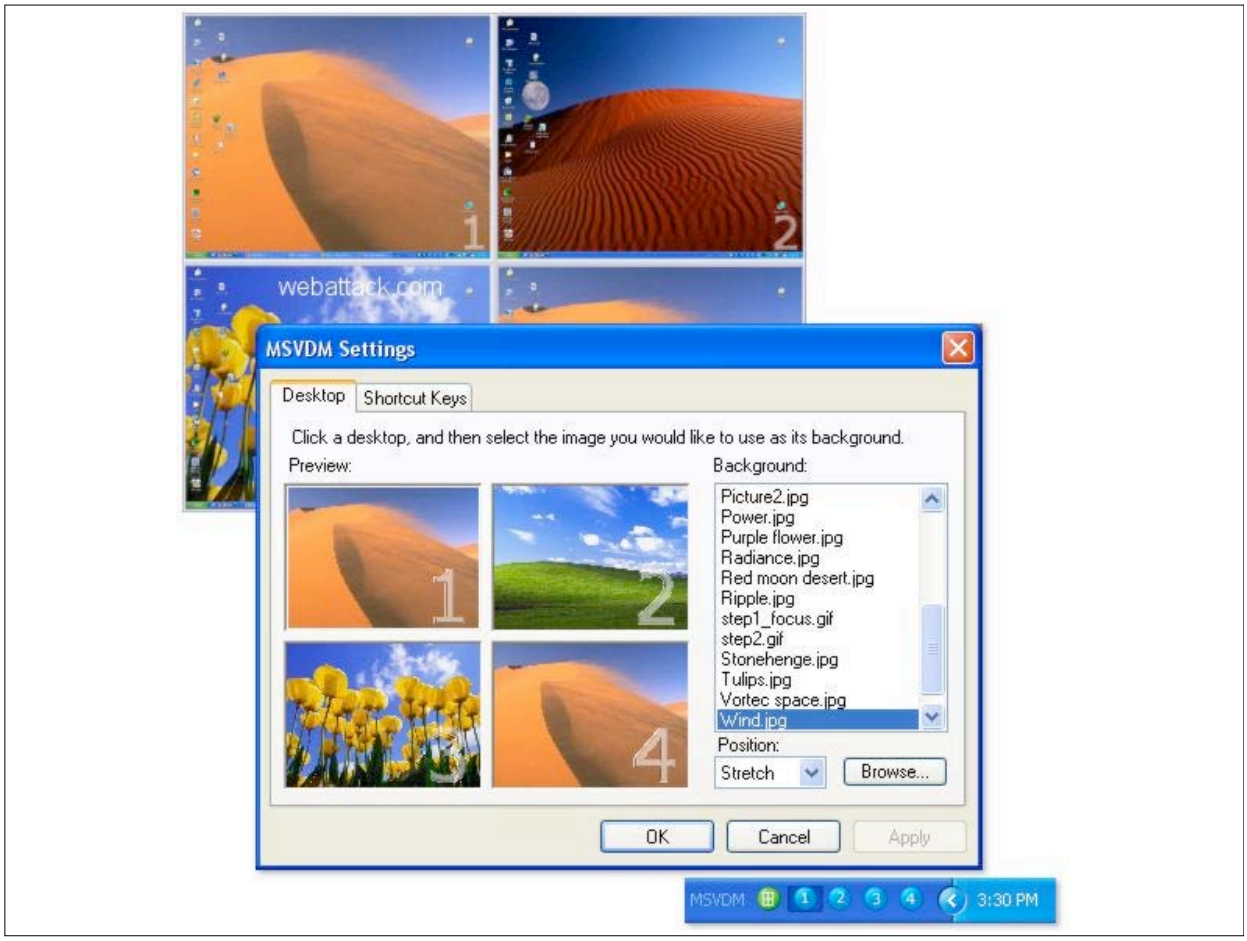
virtual desktops allow you multiple isolated environments

bind applications to desktops

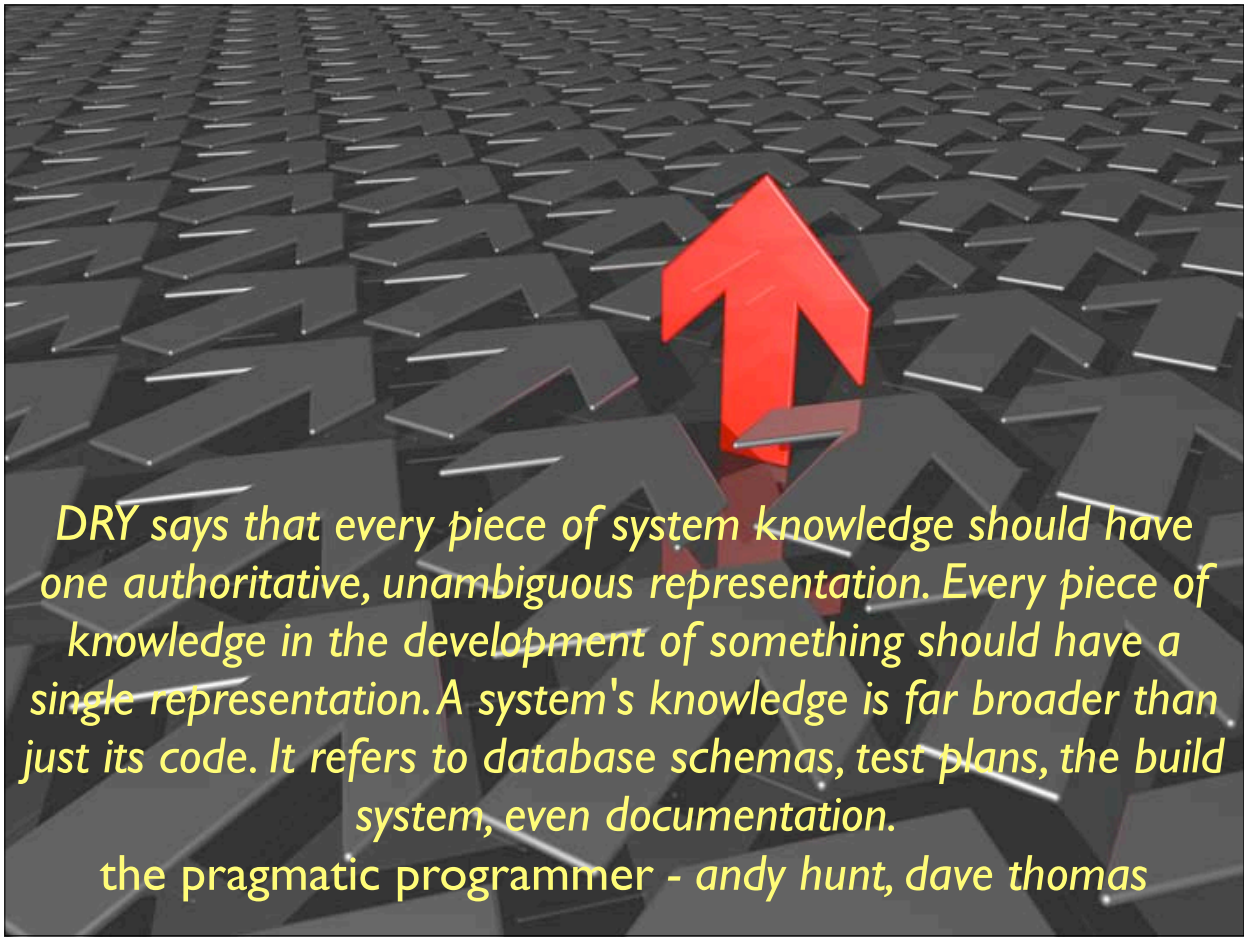
virtual desktop manager power toy

virtue desktops (in tiger)

spaces (in leopard)


webattack.com
MSVDM Settings
Desktop
Shortcut Keys
Click a desktop, and then select the image you would like to use as its background.
Preview:
Background:
Picture2.jpg
Power.jpg
Purple flower.jpg
Radiance.jpg
Red moon desert.jpg
Ripple.jpg
step1_focus.gif
step2.gif
Stonehenge.jpg
Tulips.jpg
Vortec space.jpg
Wind.jpg
Position:
Stretch
Browse...
OK
Cancel
Apply
MSVDM
3:30 PM


canonicality



dry o/r

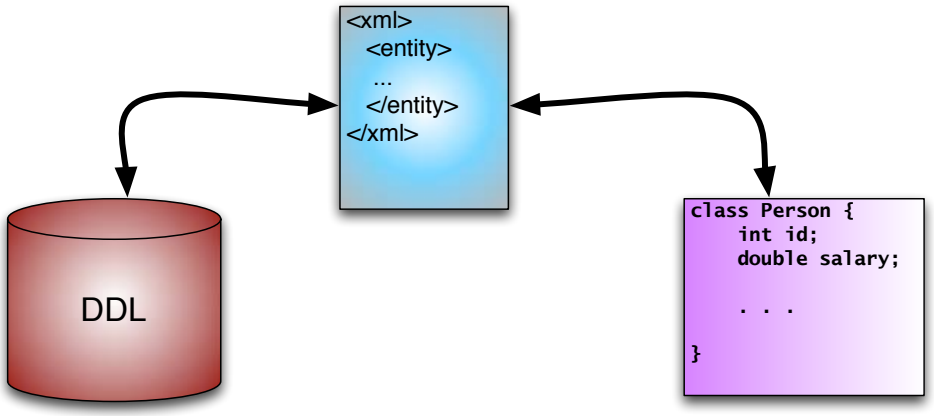
object-relational mapping is one of the most common dry violations

database schema + xml configuration + pojo > 1

decide on the canonical representation

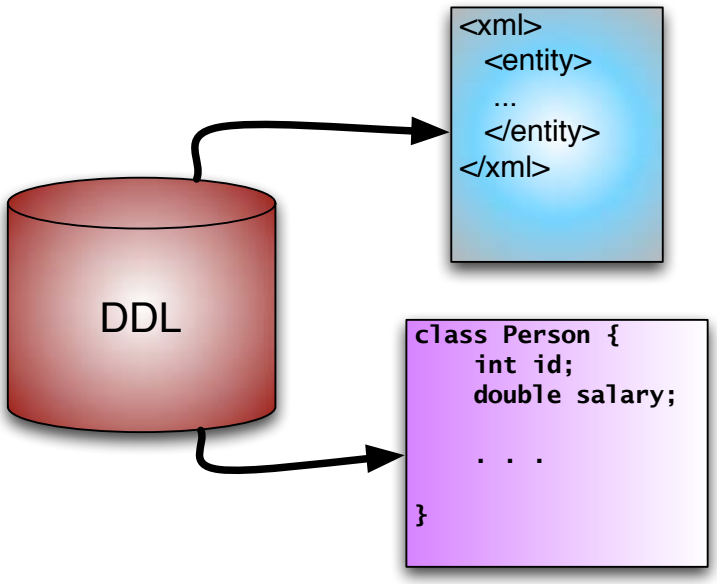
generate the others

the scenario



where's the information?

canonical representation



the target

```
<sqlMap namespace='event'>
  <typeAlias type='com.nealford.conf.canonicality.Event' alias='Event' />
  <resultMap id='eventResult' class='Event'>
    <result property='description' column='DESCRIPTION' />
    <result property='eventKey' column='EVENT_KEY' />
    <result property='start' column='START' />
    <result property='eventType' column='EVENT_TYPE' />
    <result property='duration' column='DURATION' />
  </resultMap>
  <select resultMap='eventResult' id='getEvents'>select * from event where id = ?</select>
  <select resultClass='com.nealford.conf.canonicality.Event' id='getEvent'>
      select * from event where id = #value#
  </select>
</sqlMap>
```

build event sql map

```
class GenerateEventSqlMap {
    static final SQL =
        ["sqlUrl":"jdbc:derby:/Users/jNf/work/derby_data/schedule",
            "driverClass":"org.apache.derby.jdbc.EmbeddedDriver"]
    def _file_name
    def types = [:]

    def GenerateEventSqlMap(file_name) {
        _file_name = file_name
    }
```

```
def columnNames() {
    Class.forName(SQL["driverClass"])
    def rs = DriverManager.getConnection(SQL["sqlUrl"]).createStatement().
        executeQuery("select * from event where 1=0")

    def rsmd = rs.getMetaData()
    def columns = []
    for (index in 1..rsmd.getColumnCount()) {
        columns << rsmd.getColumnName(index)
        types.put(camelize(rsmd.getColumnName(index)),
            rsmd.getColumnTypeName(index))
    }
    return columns
}
```

```
def create_mapping_file() {
    def writer = new StringWriter()
    def xml = new MarkupBuilder(writer)
    xml.sqlMap(namespace:'event') {
        typeAlias(alias:'Event',
            type:'com.nealford.conf.canonicality.Event')
        resultMap(id:'eventResult', class:'Event') {
            columnMap().each() {key, value ->
                result(property:"${key}", column:"${value}")
        }}
        select(id:'getEvents', resultMap:'eventResult',
                'select * from event where id = ?')
        select(id:"getEvent",
            resultClass:"com.nealford.conf.canonicality.Event",
                "select * from event where id = #value#")
    }

    new File(_file_name).withWriter { w ->
        w.writeLine("${writer.toString()}")
    }
}
```

generated sql map

```
<sqlMap namespace='event'>
  <typeAlias type='com.nealford.conf.canonicality.Event' alias='Event' />
  <resultMap id='eventResult' class='Event'>
    <result property='description' column='DESCRIPTION' />
    <result property='eventKey' column='EVENT_KEY' />
    <result property='start' column='START' />
    <result property='eventType' column='EVENT_TYPE' />
    <result property='duration' column='DURATION' />
  </resultMap>
  <select resultMap='eventResult' id='getEvents'>select * from event where id = ?</select>
  <select resultClass='com.nealford.conf.canonicality.Event' id='getEvent'>
      select * from event where id = #value#
  </select>
</sqlMap>
```

step 2: class builder

```
class ClassBuilder {
    def imports = []
    def fields = [:]
    def file_name
    def package_name

    def ClassBuilder(imports, fields, file_name, package_name) {
        this.imports = imports
        this.fields = fields
        this.file_name = file_name
        this.package_name = package_name
    }

    def write_imports(w) {
        imports.each { i ->
            w.writeLine("import ${i};")
        }
        w.writeLine("")
    }
```

```
def write_classname(w) {
    def class_name_with_extension = file_name.substring(
        file_name.lastIndexOf("/") + 1, file_name.length());
    w.writeLine("public class " +
        class_name_with_extension.substring(0,
            class_name_with_extension.length() - 5) + " {")
}

def write_fields(w) {
    fields.each { name, type ->
        w.writeLine("\t${type} ${name};");
    }
    w.writeLine("")
}
```

```
public class Event {
    String description;
    int eventKey;
    String start;
    int eventType;
    int duration;
```

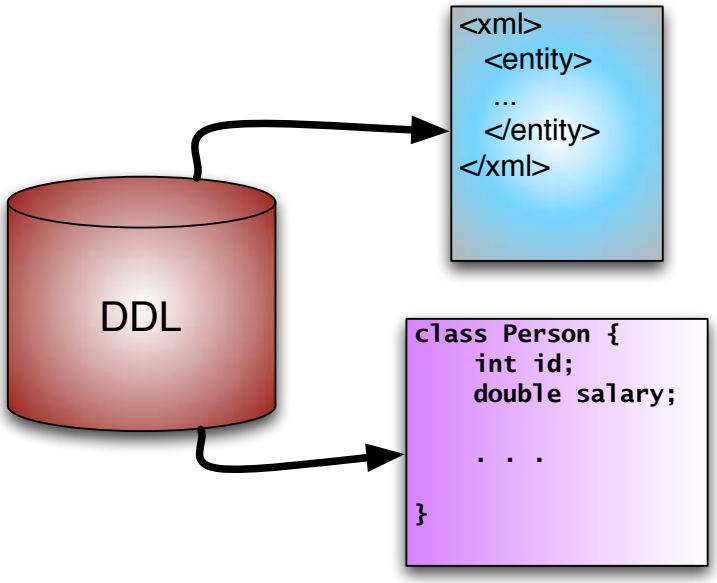
```
def write_properties(w) {
    fields.each { name, type ->
        def cap_name = name.charAt(0).toString().toUpperCase() +
            name.substring(1)
        w.writeLine("\tpublic ${type} get${cap_name}() {")
        w.writeLine("\t\treturn ${name};\n\t}\n");

        w.writeLine("\tpublic void set${cap_name}(${type} ${name}) {")
        w.writeLine("\t\tthis.${name} = ${name};\n\t}\n")
    }
}
```

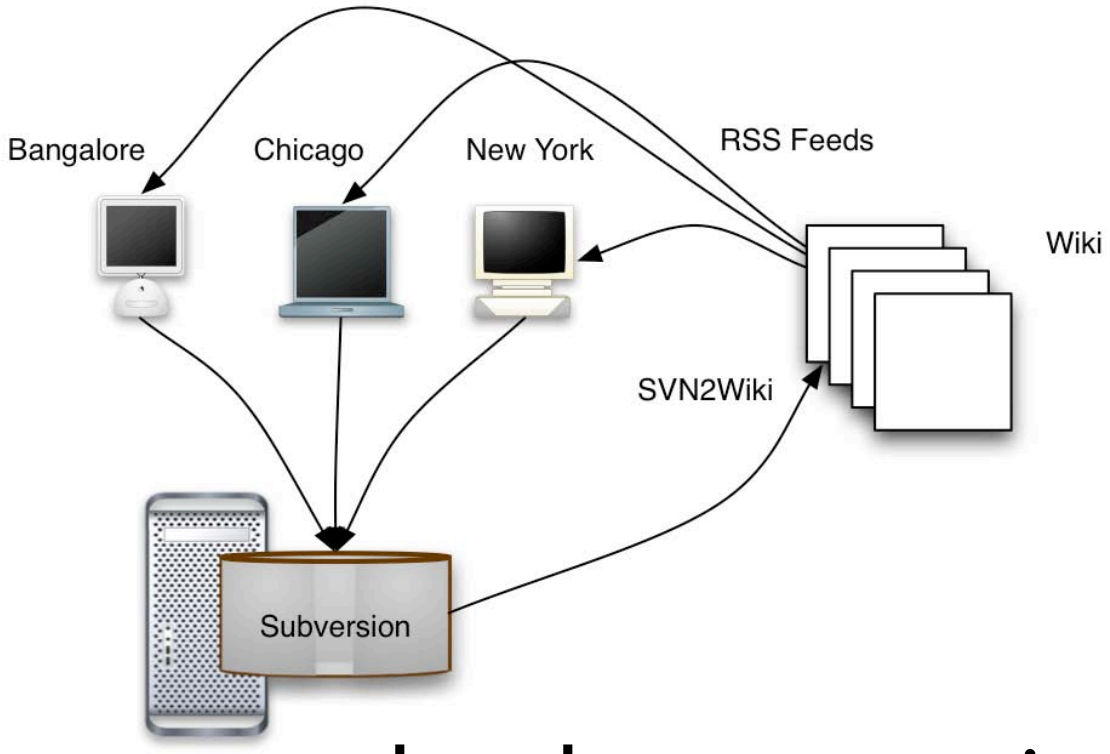
```
    return description;
}

public void setDescription(String description) {
    this.description = description;
}

// . . .
```

```
def generate_class_file() {
    new File(file_name).withWriter { w ->
        w.writeLine("package ${package_name};\n")
        write_imports(w)
        write_classname(w)
        write_fields(w)
        write_properties(w)
        w.writeLine("}")
    }
}
```

```
public class Event {
    String description;
    int eventKey;
    String start;
    int eventType;
    int duration;

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public int getEventKey() {
        return eventKey;
    }

    public void setEventKey(int eventKey) {
```

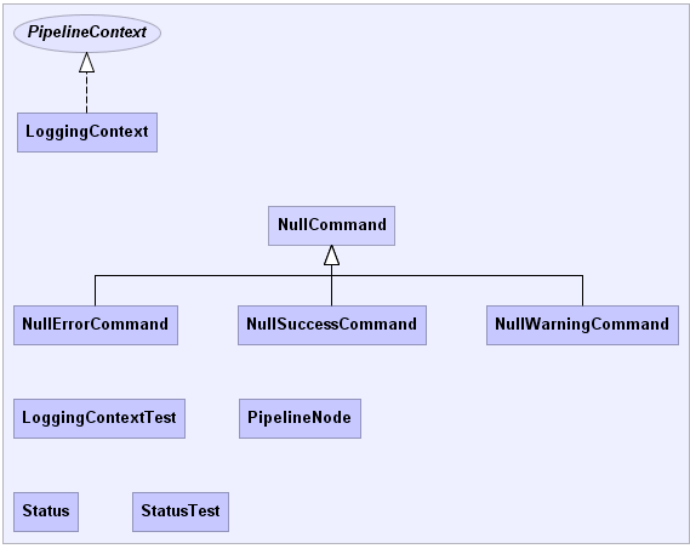
canonical representation





dry documentation

dry diagrams



dry schemas

the requirement: entity-relationship diagrams for each iteration

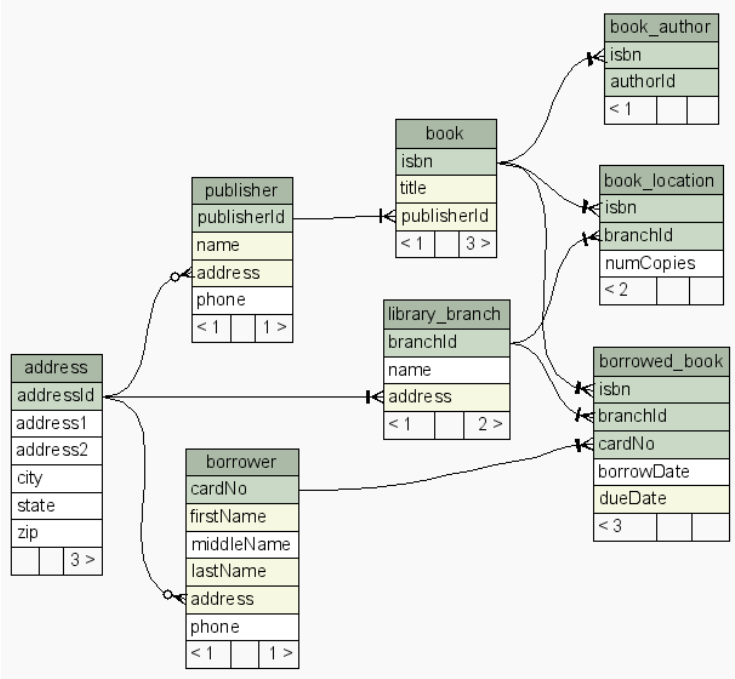
schemaspy

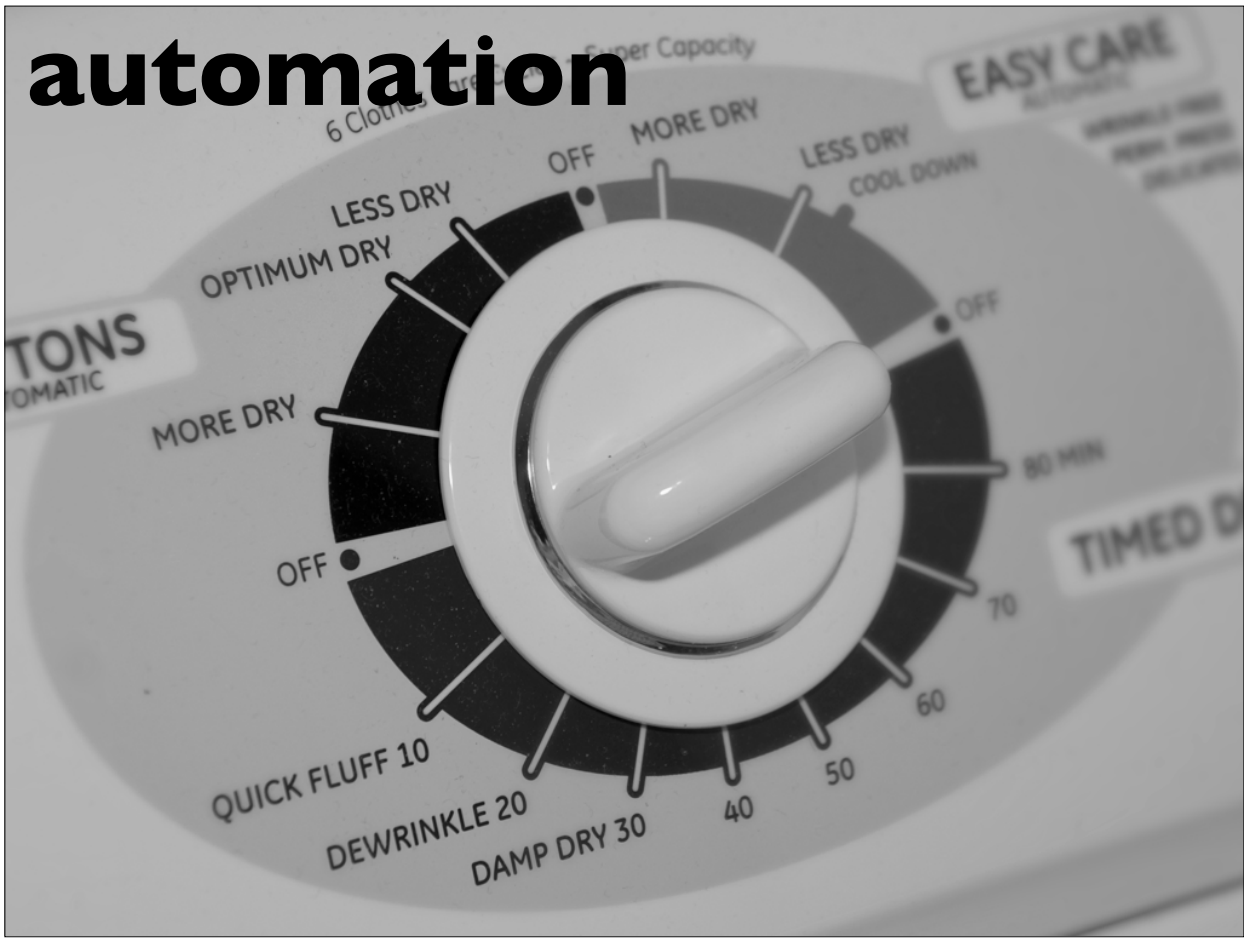
open source schema diagrammer

written in java

generates acceptable html

dry schemas


book_author
isbn
authorId
< 1
book
isbn
title
publisherId
< 1
3 >
book_location
isbn
branchId
numCopies
< 2
publisher
publisherId
name
address
phone
< 1
1 >
library_branch
branchId
name
address
< 1
2 >
borrowed_book
isbn
branchId
cardNo
borrowDate
dueDate
< 3
address
addressId
address1
address2
city
state
zip
3 >
borrower
cardNo
firstName
middleName
lastName
address
phone
< 1
1 >


automation
EASY CARE
OFF
MORE DRY
LESS DRY
COOL DOWN
LESS DRY
OPTIMUM DRY
MORE DRY
OFF
OFF
80 MIN
70
60
50
40
QUICK FLUFF 10
DEWRINKLE 20
DAMP DRY 30

obvious automatables

one-command build

continuous integration

version control (!)

documentation

don’t build what you don’t have to build

buildix

open source project from **Thought**Works

infrastructure in a box

live cd or ubuntu installation

buildix parts

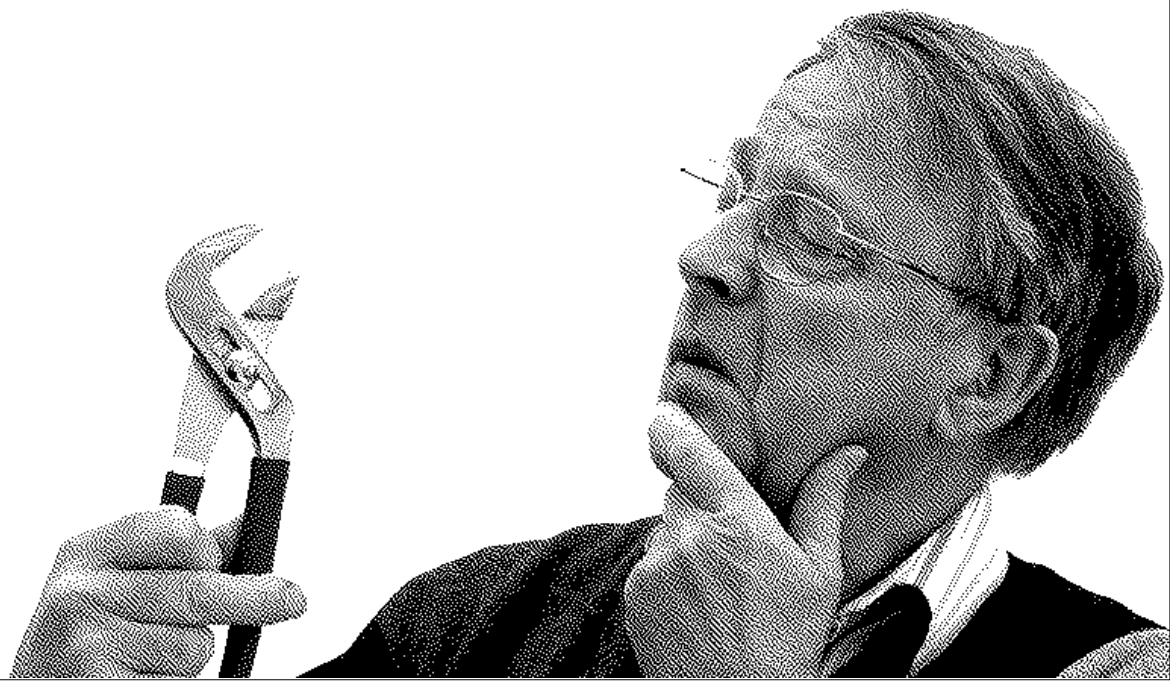
subversion

cruisecontrol

trac

mingle

subverting other tools



selenium

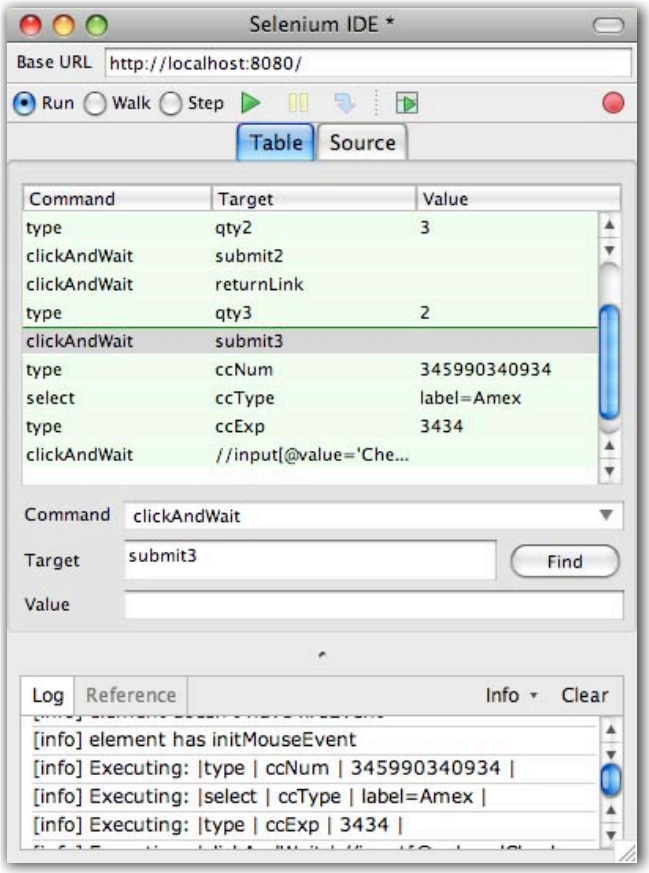
open source tool for user acceptance testing of web applications

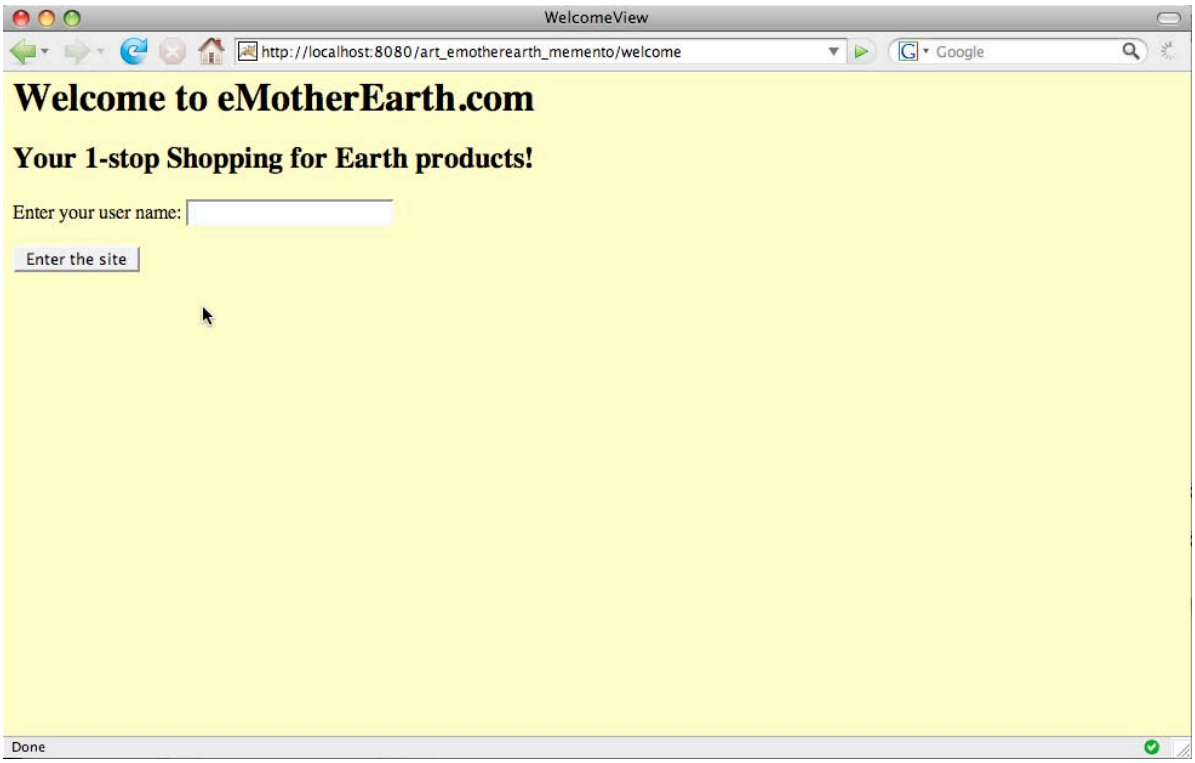
includes a side-project called selenium ide

allows you to automate debugging "wizard"-style web applications

you always think "this is the last time"...

...but it never is!


Selenium IDE *
Base URL http://localhost:8080/
Run Walk Step
Table Source
Command Target Value
type qty2 3
clickAndWait submit2
clickAndWait returnLink
type qty3 2
clickAndWait submit3
type ccNum 345990340934
select ccType label=Amex
type ccExp 3434
clickAndWait //input[@value='Che...
Command clickAndWait
Target submit3
Find
Value
Log Reference Info Clear
[info] element has initMouseEvent
[info] Executing: |type | ccNum | 345990340934 |
[info] Executing: |select | ccType | label=Amex |
[info] Executing: |type | ccExp | 3434 |


WelcomeView
http://localhost:8080/art_emotherearth_memento/welcome
Google
Welcome to eMotherEarth.com
Your 1-stop Shopping for Earth products!
Enter your user name:
Enter the site
Done

New Test		
open	/art_emotherearth_memento/welcome	
type	userName	Homer
clickAndWait	submitButton	
type	qty2	3
clickAndWait	submit2	
clickAndWait	returnLink	
type	qty6	4
clickAndWait	submit6	
type	ccNum	234234234234
select	ccType	label=MC
type	ccExp	2323
clickAndWait	//input[@value='Check out']	

```
public class NewTest extends SeleneseTestCase {
    public void testNew() throws Exception {
        selenium.open("/art_emotherearth_memento/welcome");
        selenium.type("userName", "Homer");
        selenium.click("submitButton");
        selenium.waitForPageToLoad("30000");
        selenium.type("qty2", "3");
        selenium.click("submit2");
        selenium.waitForPageToLoad("30000");
        selenium.click("returnLink");
        selenium.waitForPageToLoad("30000");
        selenium.type("qty6", "4");
        selenium.click("submit6");
        selenium.waitForPageToLoad("30000");
        selenium.type("ccNum", "234234234234");
        selenium.select("ccType", "label=MC");
        selenium.type("ccExp", "2323");
        selenium.click("//input[@value='Check out']");
        selenium.waitForPageToLoad("30000");
    }
}
```

```ruby
class NewTest < Test::Unit::TestCase
  def setup
    @verification_errors = []
    if $selenium
      @selenium = $selenium
    else
      @selenium = Selenium::SeleneseInterpreter.new(
          "localhost", 4444, "*firefox", "http://localhost:4444", 10000);
      @selenium.start
    end
    @selenium.set_context("test_new", "info")
  end

  def teardown
    @selenium.stop unless $selenium
    assert_equal [], @verification_errors
  end

  def test_new
    @selenium.open "/art_emotherearth_memento/welcome"
    @selenium.type "userName", "Homer"
    @selenium.click "submitButton"
    @selenium.wait_for_page_to_load "30000"
    @selenium.type "qty2", "3"
    @selenium.click "submit2"
    @selenium.wait_for_page_to_load "30000"
    @selenium.click "returnLink"
    @selenium.wait_for_page_to_load "30000"
    @selenium.type "qty6", "4"
    @selenium.click "submit6"
    @selenium.wait_for_page_to_load "30000"
    @selenium.type "ccNum", "234234234234"
    @selenium.select "ccType", "label=MC"
    @selenium.type "ccExp", "2323"
    @selenium.click "//input[@value='Check out']"
    @selenium.wait_for_page_to_load "30000"
  end
end
```

automated interaction

record your interaction the 1st time you walk through the page

literally cuts hours off debugging time

selenium defines an interaction api for web applications

have your q/a department record bug discoveries

don’t spend time doing by
hand what you can automate

build your own tools

you almost never do anything just once

work like a craftsman, not a laborer

build shims & jigs

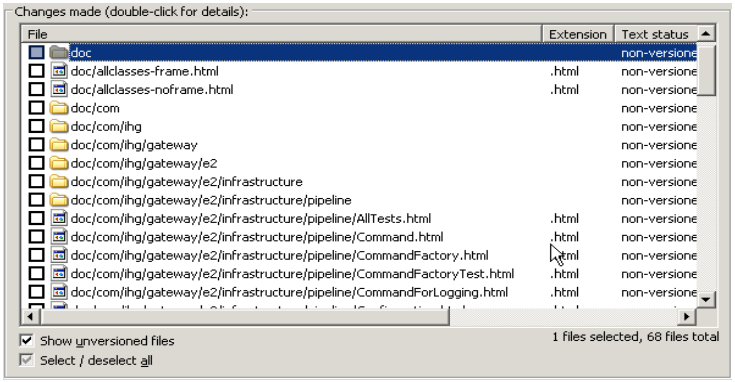
building a tool takes a little longer than brute force...

...but you build assets

bash-fu

adding new files to subversion repository

tortoise (on windows), but with limits



svnAddNew

```
svn st | grep '^\?' | tr '^\?' ' ' |
sed 's/ [ ]*//' | sed 's/[ ]/\\ /g' | xargs svn add
```

`svn st`	get svn status (new files start with "?")
`grep '^\?'`	find all new files
`tr '^\?' ' '`	translate the "?" into a space
`sed 's/[ ]*//'`	substitute spaces to nothing
`sed 's/[ ]/\\ /g'`	escape embedded spaces
`xargs svn add`	pipe the improved arguments into svn

more bash-fu

the problem: 2 gb of log files / week

need to know the count of each exception type

by hand?!?

automate with a bash script

get all exception types from log file

sort them

```
#!/bin/bash
for X in $(egrep -o "[A-Z]\w*Exception" genesis_week.txt | sort | uniq) ;
do
    echo -n -e "$X\t"
    grep -c "$X" genesis_week.txt
done
```

get unique list

get counts of each exception

automating com

```
def open_daily_logs
  excel = WIN32OLE.new("excel.application")

  workbooks = excel.WorkBooks
  excel.Visible = true
  doc_list.each do |f|
    begin
      workbooks.Open(@@Home_Dir + f, true)
    rescue
      puts "Cannot open workbook:", @@Home_Dir + f
    end
  end
  excel.Windows.Arrange(7)
end
```

scripting rationale

examples in lots of different languages/tools

which one do I use for this problem?

use a *real* language for scripting

sql splitter

the problem: split a 38,000 line sql file into 1000 line chunks

each chunk must be syntactically correct

“we can do it by hand in 10 minutes...”

automate instead

after 50 minutes:

```
SQL_FILE = "./GeneratedTestData.sql"
OUTPUT_PATH = "./chunks of sql/"

line_num = 1
file_num = 0
Dir.mkdir(OUTPUT_PATH) unless File.exists? OUTPUT_PATH
file = File.new(OUTPUT_PATH + "chunk " + file_num.to_s + ".sql",
    File::CREAT|File::TRUNC|File::RDWR, 0644)

done, seen_1k_lines = false
IO.readlines(SQL_FILE).each do |line|
  file.puts(line)
  seen_1k_lines = (line_num % 1000 == 0) unless seen_1k_lines
  line_num += 1
  done = (line.downcase =~ /^\W*go\W*$/ or
          line.downcase =~ /^\W*end\W*$/) != nil
  if done and seen_1k_lines
    file_num += 1
    file = File.new(OUTPUT_PATH + "chunk " + file_num.to_s + ".sql",
          File::CREAT|File::TRUNC|File::RDWR, 0644)
    done, seen_1k_lines = false
  end
end
```

time spent automating

it took us 5 times longer to automate it

we've had to do it numerous times since

it "accidentally" became an important part of our project

using a real language allowed us to refactor it...

...so that we could write unit tests

```ruby
def test_mocked_out_dir
  ss = SqlSplitter.new("dummy_path", "dummy_file")
  Dir.expects(:mkdir).with("dummy_path")
  ss.make_a_place_for_output_files
end

def test_that_output_directory_is_created_correctly
  ss = SqlSplitter.new(OUTPUT_PATH, nil)
  ss.make_a_place_for_output_files
  assert File.exists? OUTPUT_PATH
end

def test_that_lines_o_sql_has_lines_o_sql
  lines = %w{Lorem ipsum dolor sit amet consectetur}
  ss = SqlSplitter.new(nil, nil)
  ss.sql_lines = lines
  assert ss.lines_o_sql.size > 0
  assert_same ss.lines_o_sql, lines
end

def test_generate_sql_chunks
  ss = SqlSplitter.new(OUTPUT_PATH, nil)
  ss.sql_lines = lots_o_fake_data
  ss.generate_sql_chunks
  assert File.exists? OUTPUT_PATH
  assert Dir.entries(OUTPUT_PATH).size > 0
  Dir.entries(OUTPUT_PATH).each do |f|
    assert f.size > 0
  end
end
```

using real languages

allow throw-aways to grow into assets

allows unit testing, refactoring, ide support

if you start by treating it as a 1st class problem, you'll build better solutions

time savings

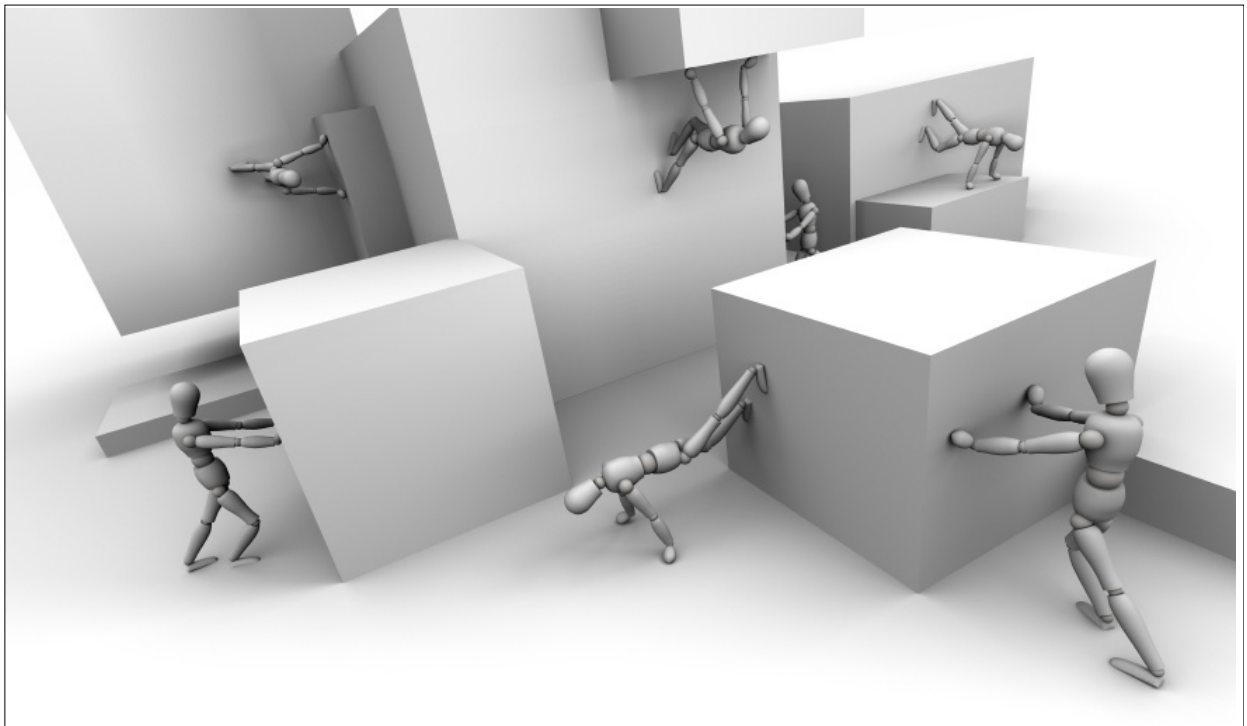
solving problems by hand makes you dumber

steals concentration

squanders focus

automating makes you smarter

figure out clever ways to solve problems



justifying automation

timebox

set a reasonable time to see if it's possible

evaluate at the end of the box

decide if you want to go forward

or create another time box

or abandon the effort

analyze the r.o.i.

how long does it take now X # of times we must do it?

what are the consequences of doing it wrong 1 time?

automation is about

- time savings
- risk mitigation

questions?

please fill out the session evaluations
slides & samples available at nealford.com

NEAL FORD software architect / meme wrangler

ThoughtWorks

nford@thoughtworks.com
3003 Summit Boulevard, Atlanta, GA 30319
www.nealford.com
www.thoughtworks.com
memeagora.blogspot.com

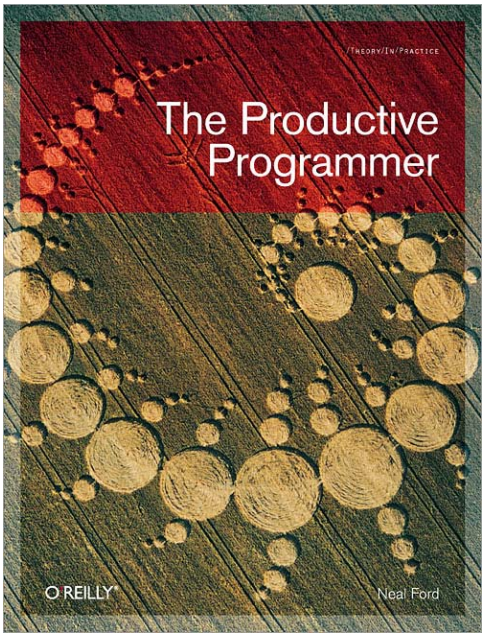
resources

The Productive Programmer

Published by O'Reilly Media

ISBN: 978-0-596-51978-0

Photos by Candy Ford



NF