Agile Engineering Practices

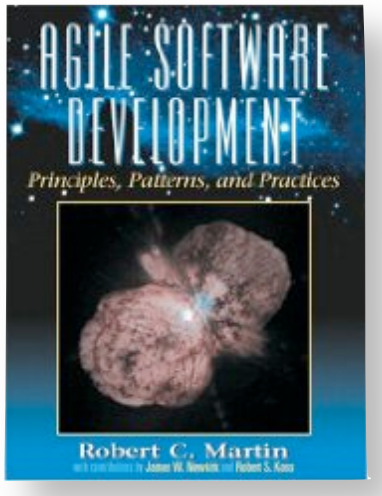
NEAL FORD software architect / meme wrangler

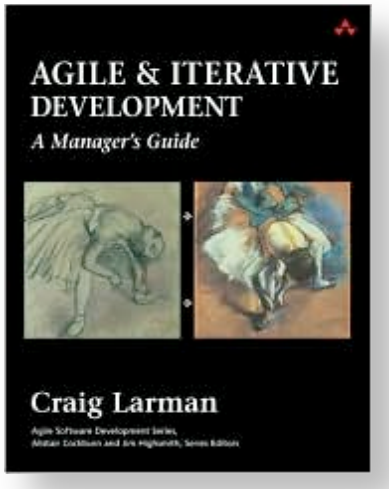
ThoughtWorks®

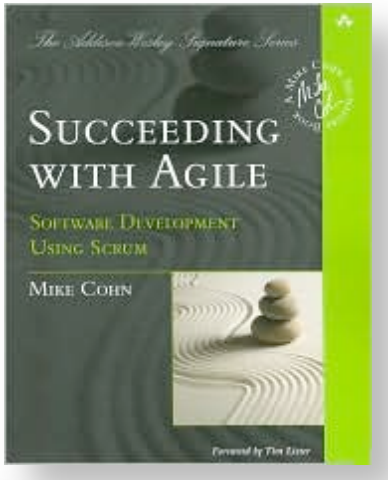
nford@thoughtworks.com
3003 Summit Boulevard, Atlanta, GA 30319
www.nealford.com
www.thoughtworks.com
blog: memeagora.blogspot.com
twitter: neal4d

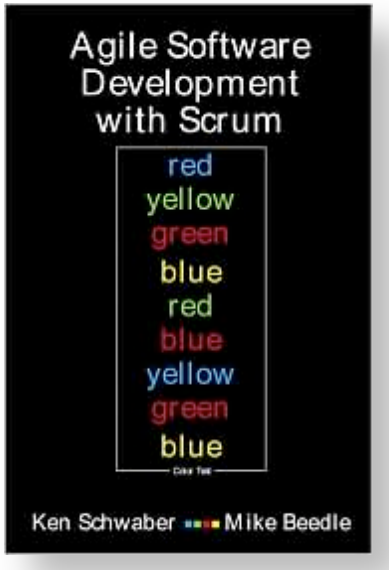
planning vs doing

Go for the one that'll beat the one you last did

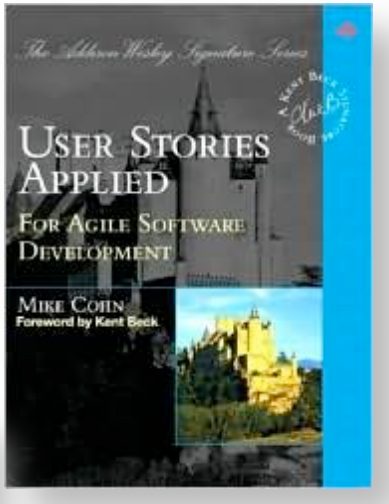

AGILE SOFTWARE DEVELOPMENT
Principles, Patterns, and Practices
Robert C. Martin

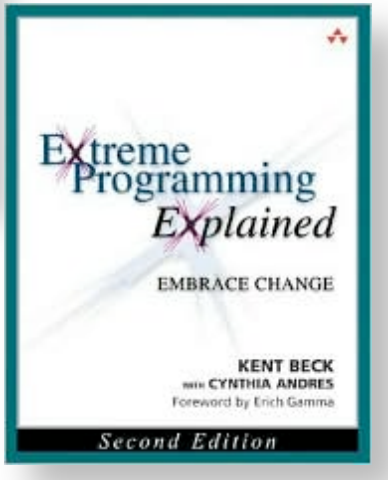

AGILE & ITERATIVE DEVELOPMENT
A Manager's Guide
Craig Larman

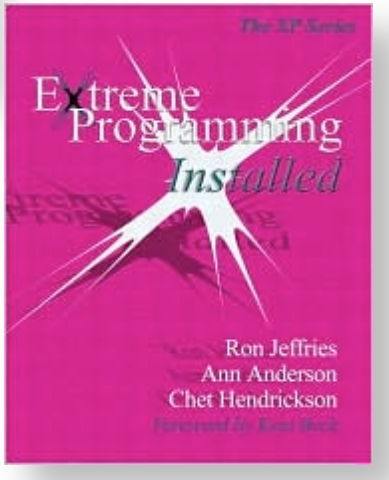

SUCCEEDING WITH AGILE
SOFTWARE DEVELOPMENT USING SCRUM
MIKE COHN

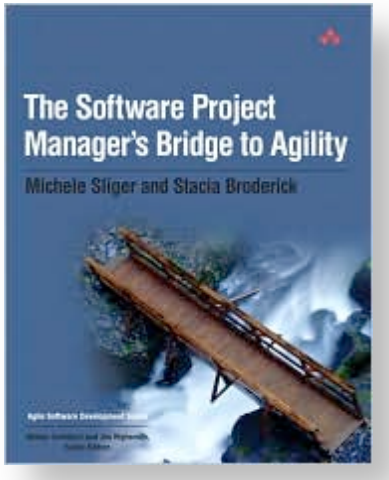

Agile Software Development with Scrum
red
yellow
green
blue
red
blue
yellow
green
blue
Ken Schwaber
Mike Beedle


Microsoft
AGILE PROJECT MANAGEMENT WITH SCRUM


USER STORIES APPLIED
FOR AGILE SOFTWARE DEVELOPMENT
MIKE COHN
Foreword by Kent Beck


Extreme Programming Explained
EMBRACE CHANGE
KENT BECK
with CYNTHIA ANDRES
Foreword by Erich Gamma
Second Edition


Extreme Programming Installed
Ron Jeffries
Ann Anderson
Chet Hendrickson

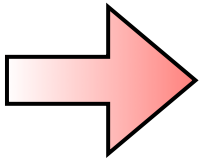

The Software Project Manager's Bridge to Agility
Michele Sliger and Stacia Broderick

planning is stage one

most agile methodologies ignore engineering

why does scrum hate developers?

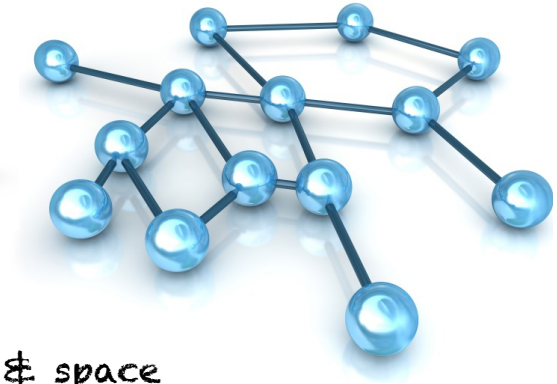
developers gone wild?!?

agility  discipline

feedback loops

automation

metrics



time & space

communication

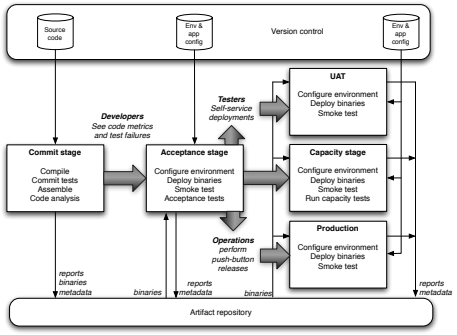
non-

intuitivity

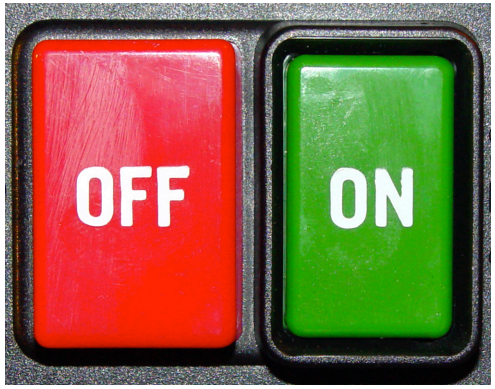
demonstration

which falls faster?

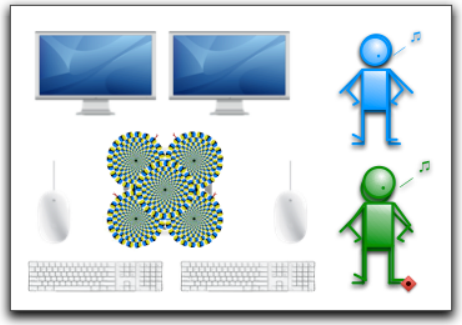
agile estimation
[15 mins]



continuous delivery
[10 mins]

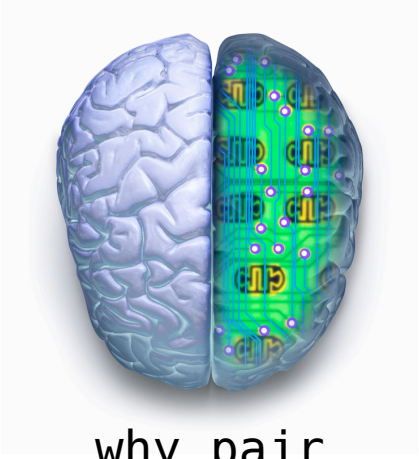


feature toggles
[10 mins]

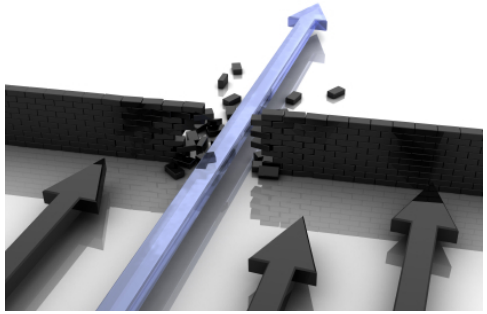


DVCS magic
[10 mins]

design practices
[15 mins]



why pair
programming works
[10 mins]



TDD & velocity
[10 mins]

pair programming
mechanics [15 mins]

Try playing the throw that would have lost to your opponents last throw.

agile

estimation

iteration 0 (inception)

architecture QoS testing

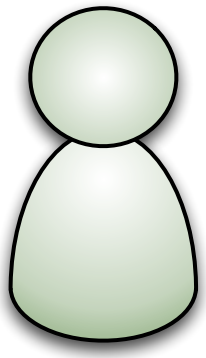
what does it do? when?

estimation

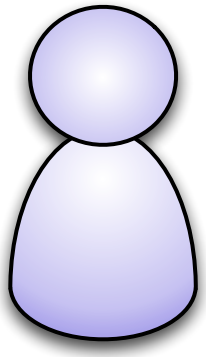
```
for each story {
    BA/stakeholder describes what
    it does

    developers gauge complexity

    assign complexity points

}
```

complexity vs. time

rookie



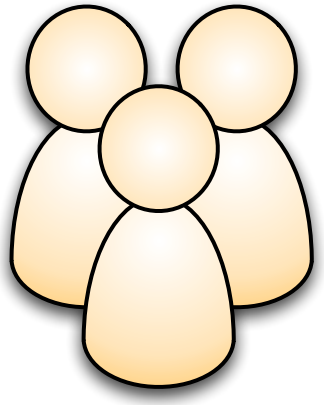
time = coding with no interruption
16 hours a day, subsisting on
cold pizza & mountain dew



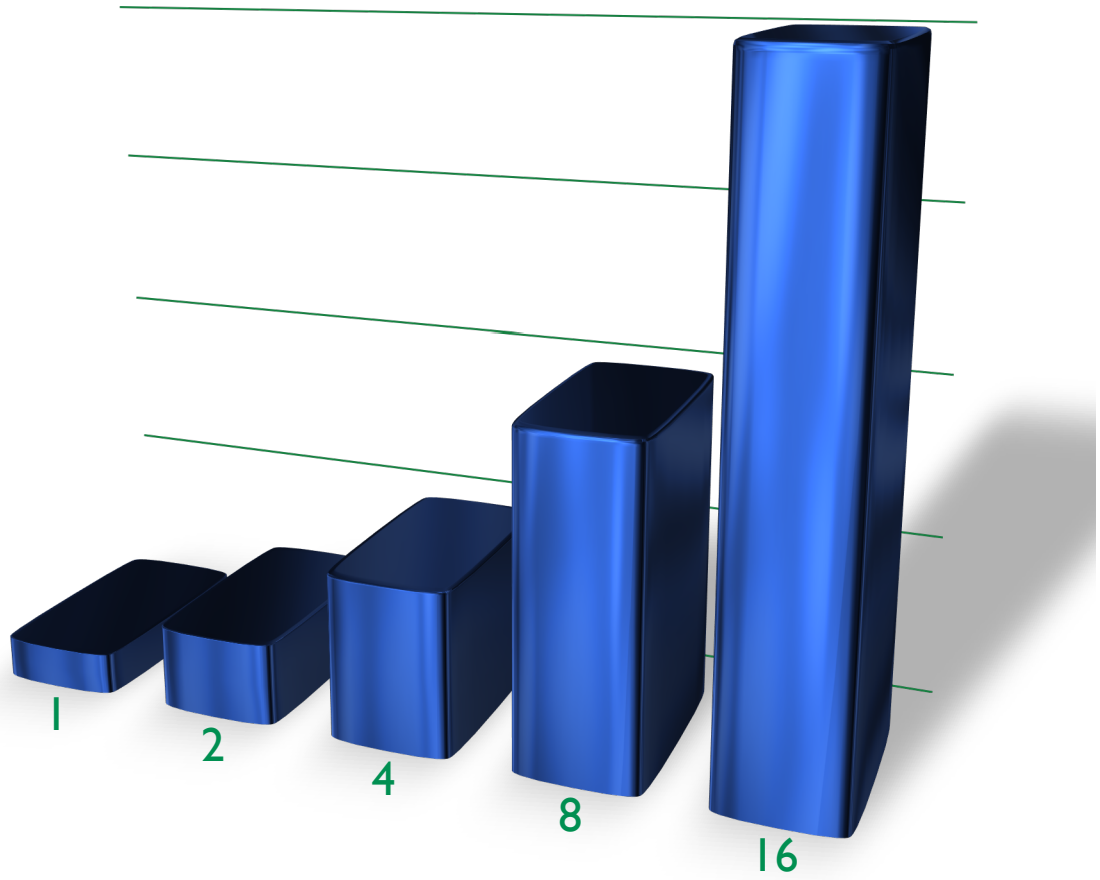
time = 8 hrs **-** (#_meetings **+**
support_calls **+**
email + fixing_printers)

experienced
developer

complexity



how complex is
this story compared
to other stories?



complexity

less ad-hoc variable values

more consistent across projects

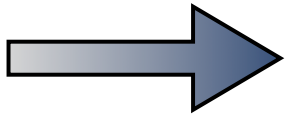
gets better over time

builds trust

project manager assigns load factor to convert complexity to time

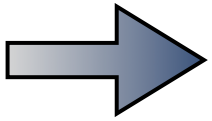

business chooses
story order

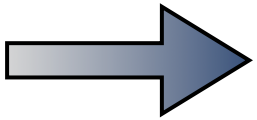
estimation & metrics

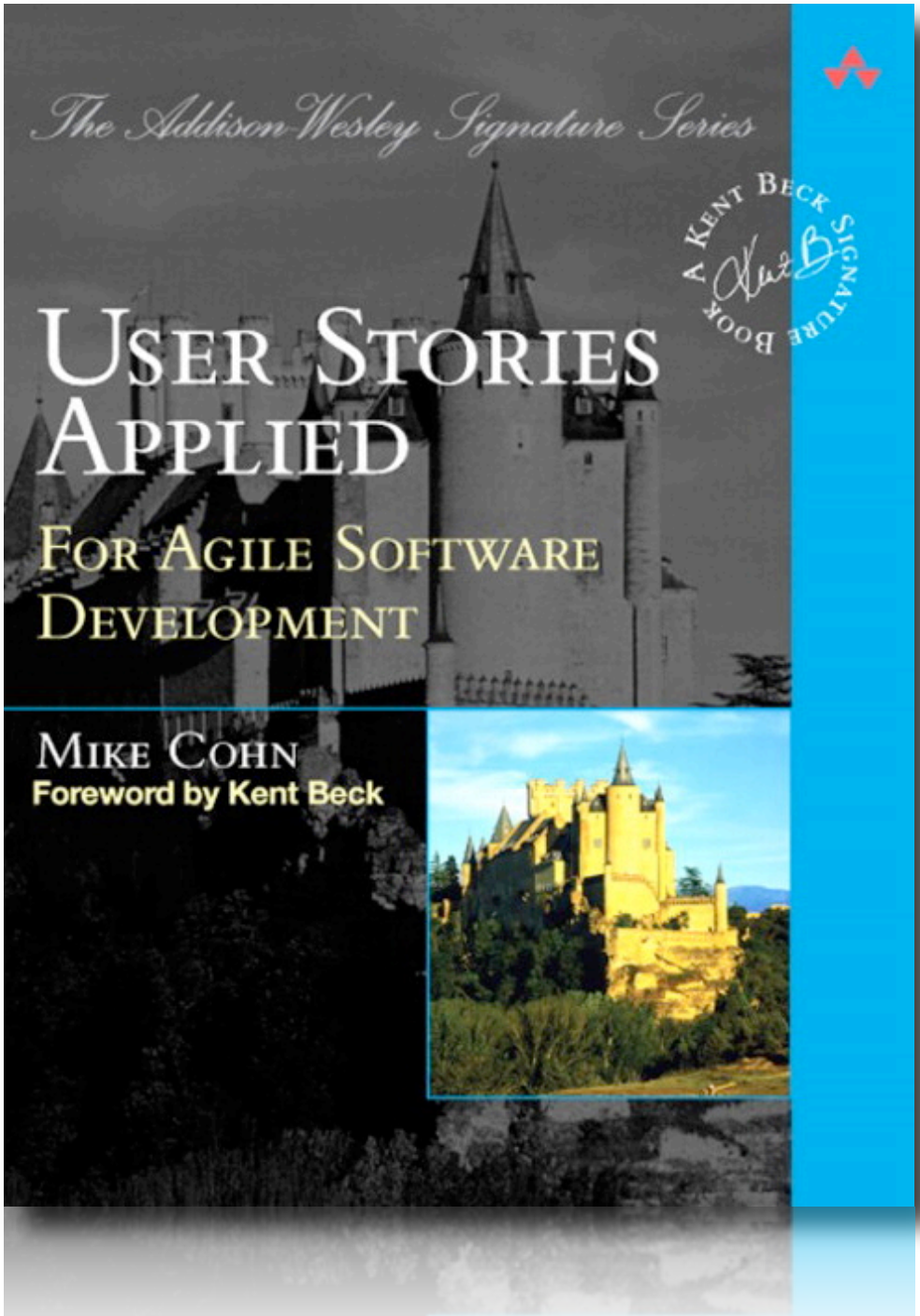
quality of data  quality of metrics

coarse grained estimate by developers is a good starting place

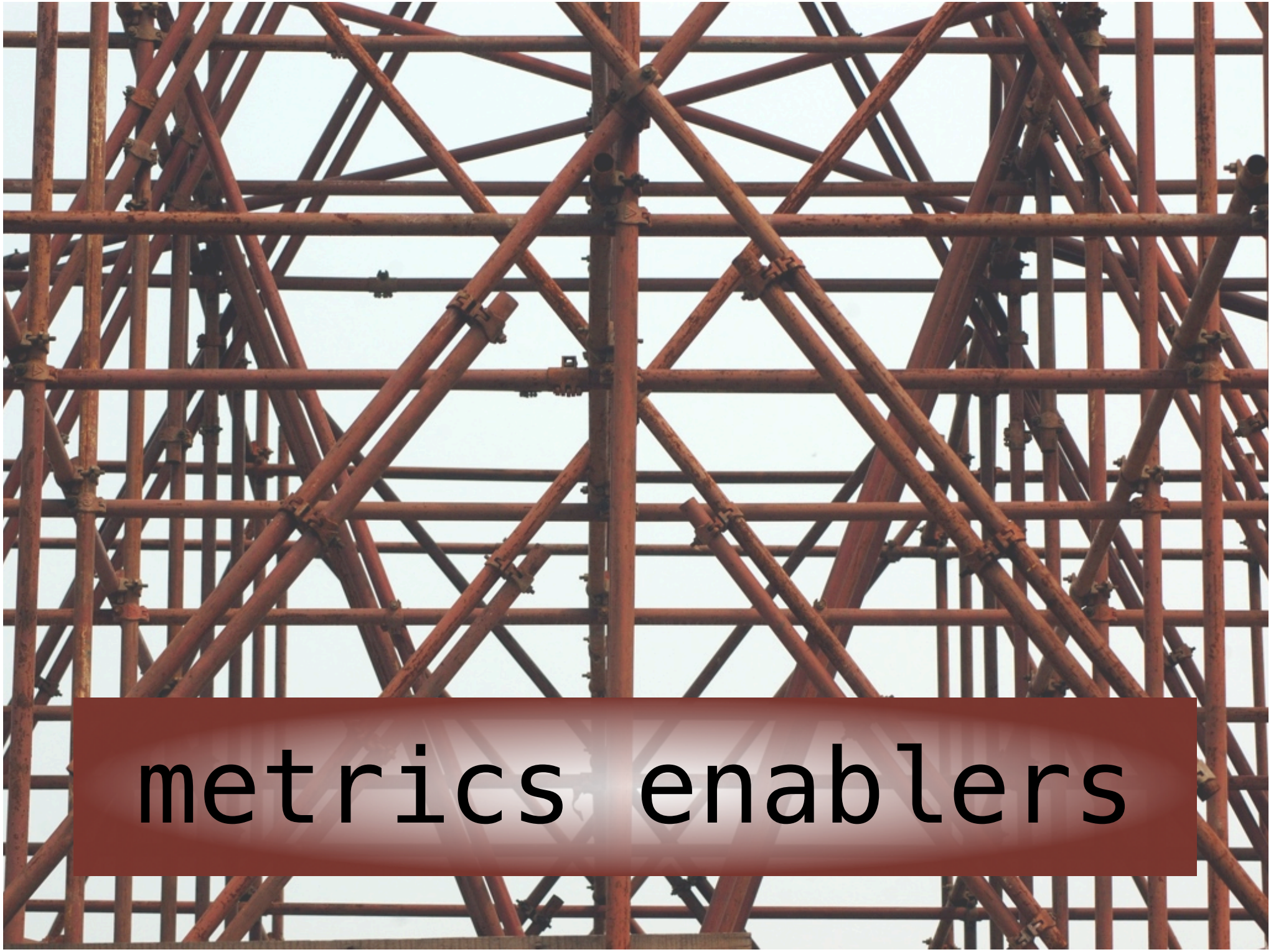
project manager continuously gauges the quality of estimates using actual data

if the load factor if wrong  change it

if the estimates are poor  re-estimate



for
more
information


metrics enablers

business derived completion criteria

feedback loops

communication

Sample Story Card Request Manager to source widgets

Status: Approved

VERSIONING & APPROVAL

Version	Author/Modifier	Date	Changes	Approval
1.0	BA Bob	2/28/2006	Original	
1.1	BA Bob	3/1/2006	updated after meeting with users	Approved

STORY

Provide sales the ability to mark as ready for review and send an email to Ops.

BUSINESS CONTEXT

When a sales manager is done with his/her work on an inquiry in terms of searching for and/or selecting widgets to it, operations needs to be informed to complete sourcing information on the order.

DEVELOPMENT RELEASE / ITERATION: RELEASE 1 - ITERATION 3

KEY PROCESS AREA: Order Inquiry

ANALYST: BA Bob

LIMITATIONS

Story Widget#	Business Segment	Story Title
OI-20	Order Inquiry	Add customer & delivery information to order
OI-47	Order Inquiry	Define user roles for Order Inquiry and Order Management

EXISTING CONDITIONS:

Creating an inquiry and including customer information on the inquiry is possible in The Widget Project (OI-36, OI-3). Saving an inquiry is possible (OI-2) as well as selection of widgets on to an inquiry (OI-5a, OI-5b). Story OI-11a has been played which is the first story to setup an email alert.

STORY NARRATIVE (including examples):

The story begins from the point where a user has created a qualified inquiry where widgets may or may not have been selected.

1. **Sales Manager is able to mark the inquiry as ready for review by operations.**

On the Inquiry screen below the search results and widget selections, display this label 'Inquiry is ready for review by Manager and a submit button.

Once the submit button has been pressed, disable the button.

2. **Associate Managers to Sales Managers**

Sales Manager	Manager
Frank Diodati	Debbie Bone
John Martin	Debbie Bone
Mark Poepping	Ellen Richter
Paul Curry	Ellen Richter
George Sullivan	Rick Leslie
Mike Keasling	Rick Leslie
Tim Ewing	Maria Dobanovacki
Craig Newlun	Maria Dobanovacki
John Glynn	Margaret Ebert
Jeff Rasmussen	Kelli Wisla
Bill Lyness	Kelli Wisla
Mike Calabucci	Laura Felix
Sonia Faucher	Alejandra Mendez

Store the names as First Name and Last Name.

3. Send alert to the associated Manager when inquiry is marked ready for review

The email alert should follow the format described in OI-11a and contain the following information:

Subject – Inquiry <inquiry name> is ready for review

<salutation> < manager name>,

Inquiry <inquiry name> for customer <customer name> created by <sales manager name> is ready for your review to complete sourcing and delivery information.

4. Once an inquiry is marked ready for review, allow only the Manager to change Inquiry criteria or widget selections if made.

Maintain a " Ops Review Request Indicator" which will indicate that the inquiry has been sent to Ops for review. Also maintain the date on which the request is made. On the Inquiry screen, display " Ops Requested on <date>".

5. Add Ops owner to Search for Inquiry criteria

VALIDATIONS (INCLUDE EXPECTATION FOR NOTIFYING USER OF INVALID INPUT)

None.

AUDIT TRAIL

All changes should be logged.

SECURITY

Create a permission for the ready for review submission button.

C a

Sample Story Card Request Manager to source widgets

Status: Approved

Impact to Other Systems - Integration

None.

Performance considerations

None.

Screen Mockup (as needed)

Not required.

User Documentation/online help Required

Not at this time.

Tests Required (include all "happy" routes)

1. On qualified inquiries, the user is presented with a label 'Inquiry is ready for review by Manager' and a submit button.
2. Once depressed, the submit button is disabled.
3. On submission, an email in the desired format is sent to the Manager associated to the Sales Manager.
4. Check that only the Manager is able to change search criteria or widget selections once marked ready for review. Check that the date request was made displays.
5. Check that the Ops owner list shows up as a criteria in Search for Inquiry

Additional Tests

[Include additional testing covering other areas of the system that may indirectly be impacted by the changes in the above narrative] QA will update this area as they create the tests if necessary

Regression Test Requirements

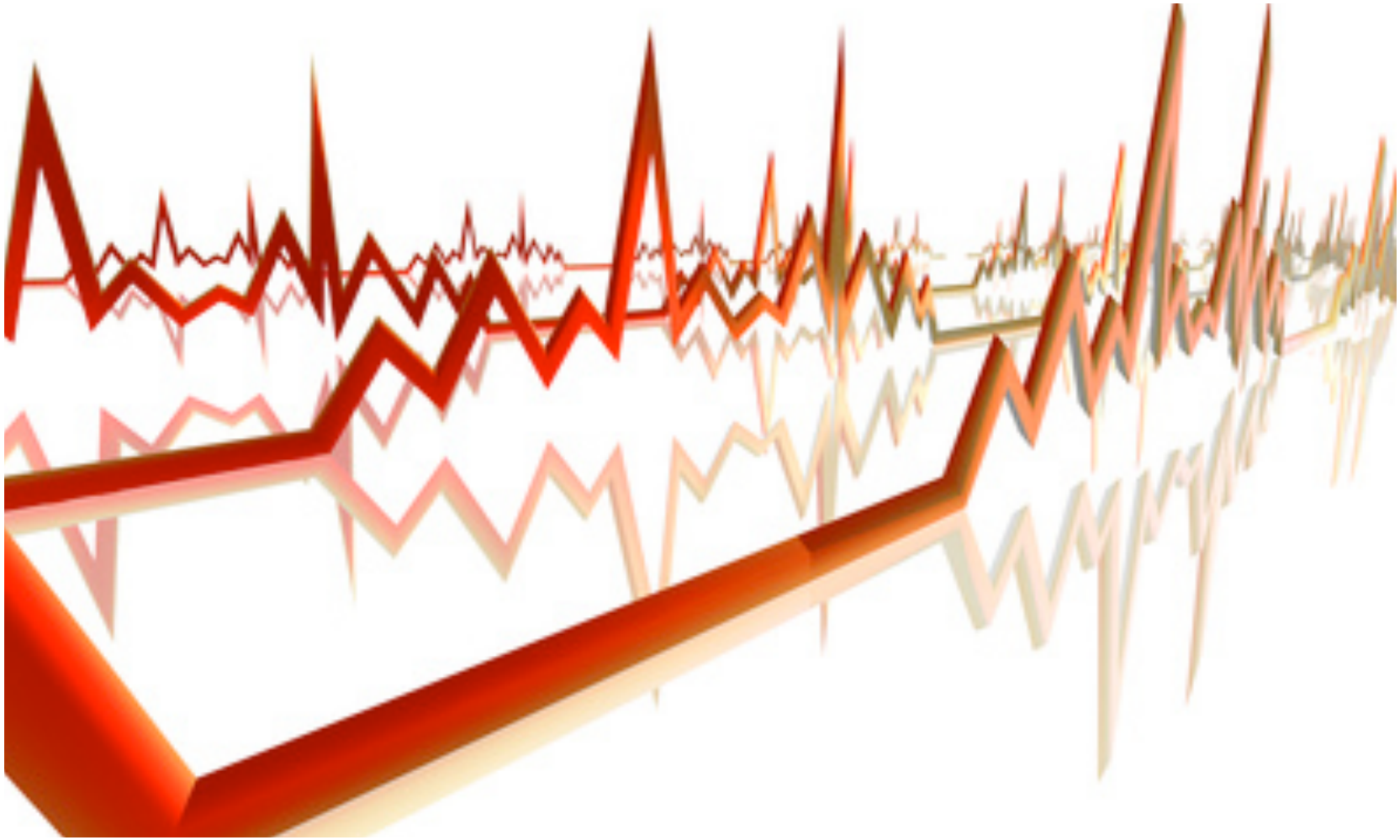
This test may be made available in the regression suite.

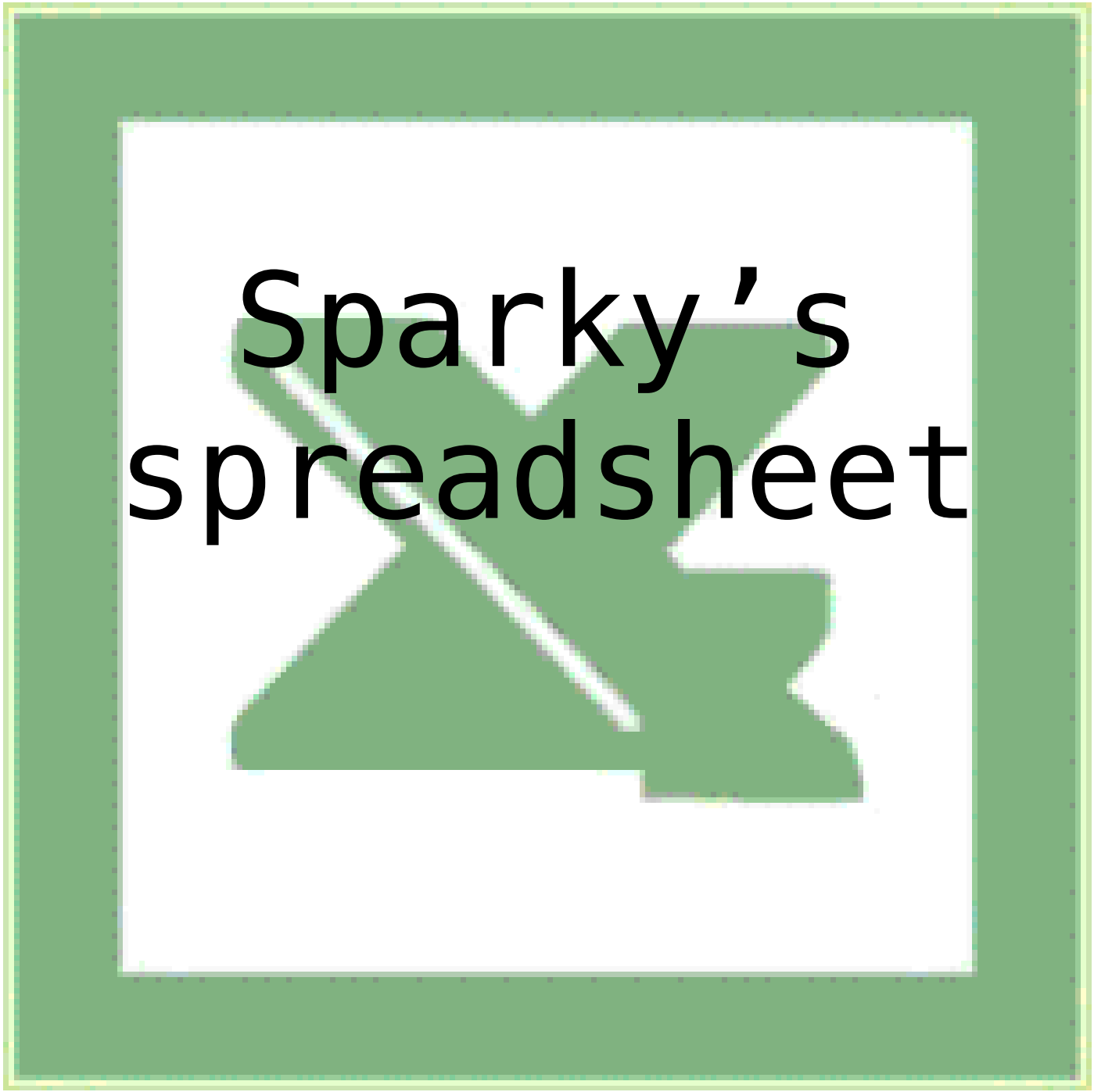
metrics

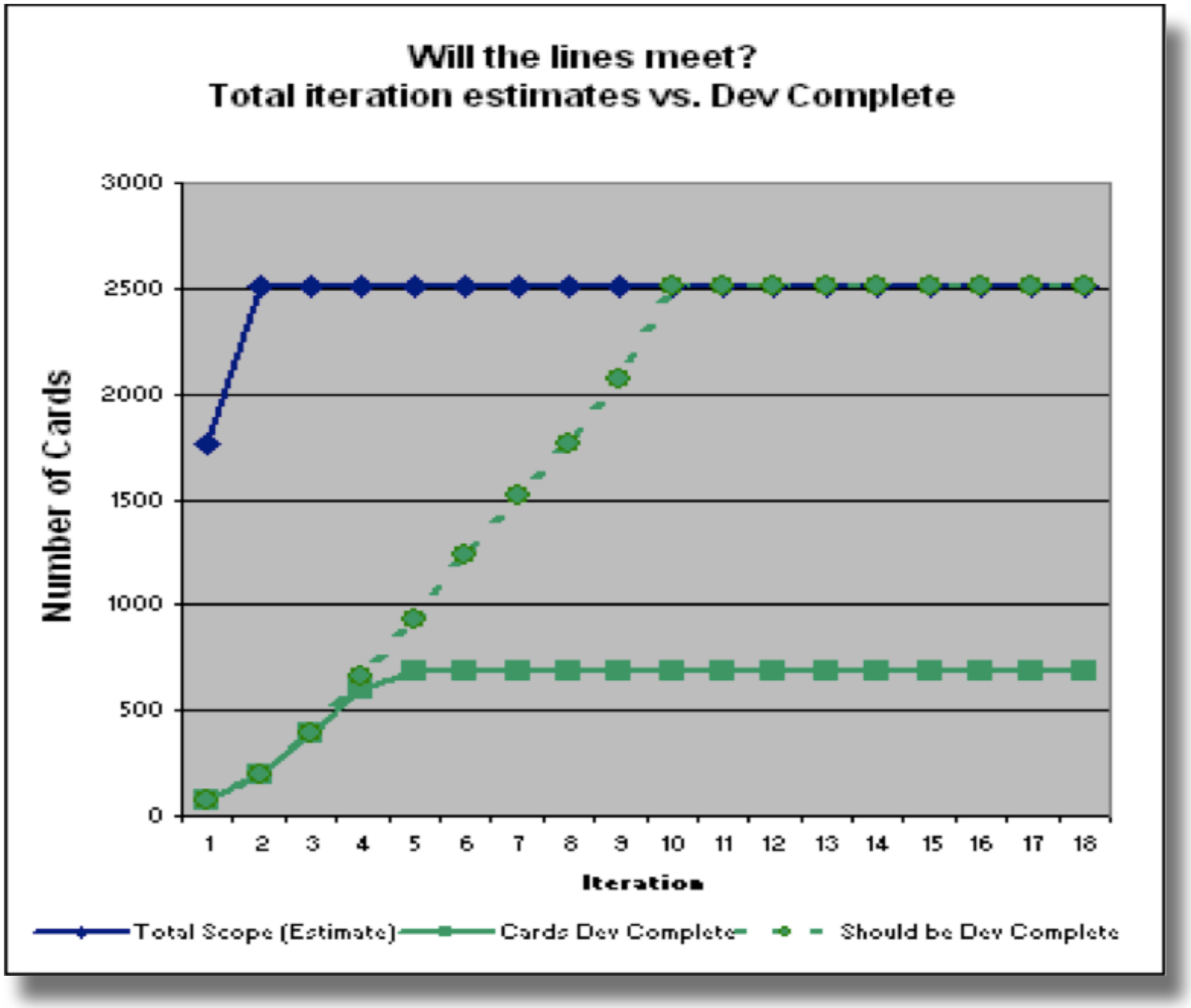
binary completion

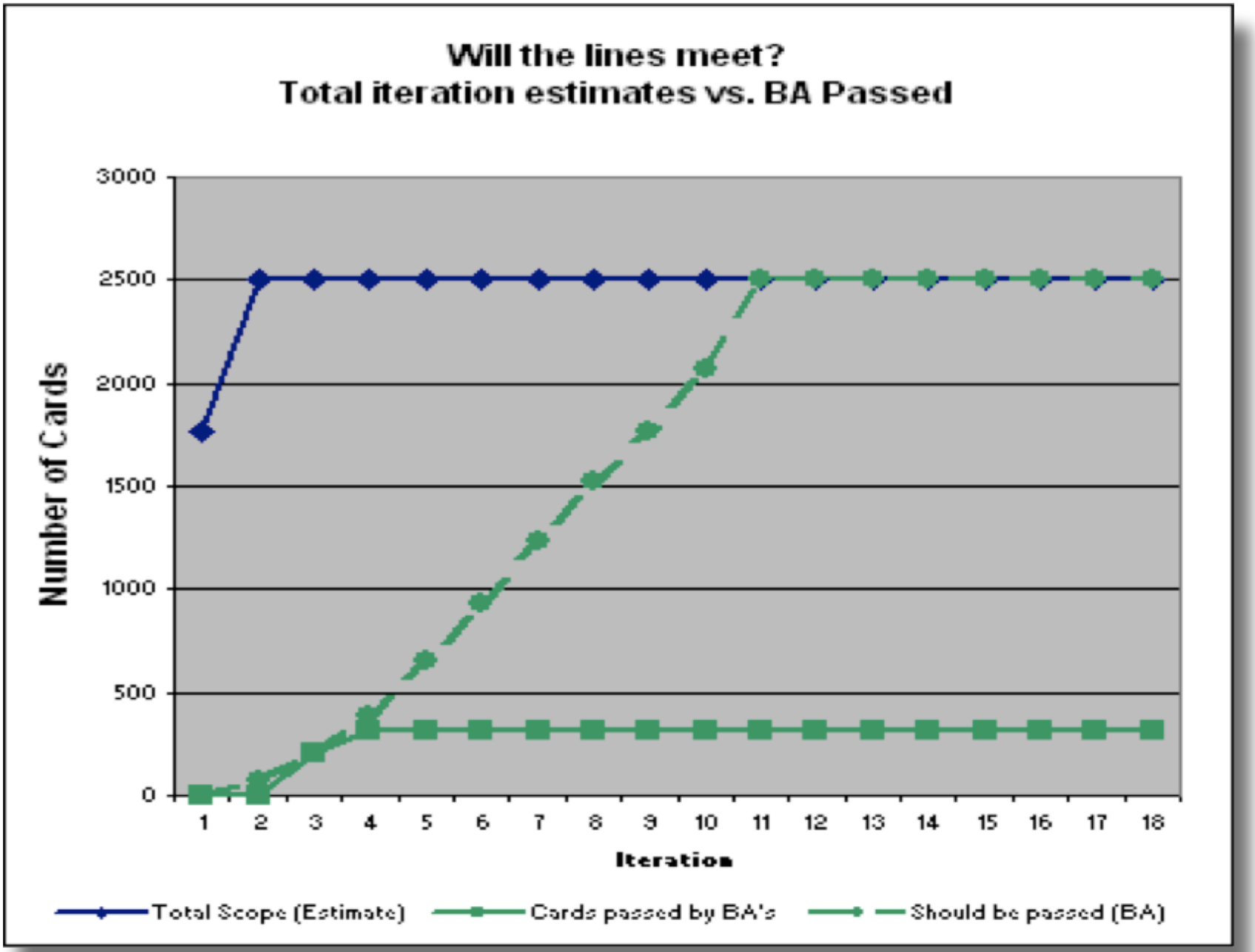
0→1.

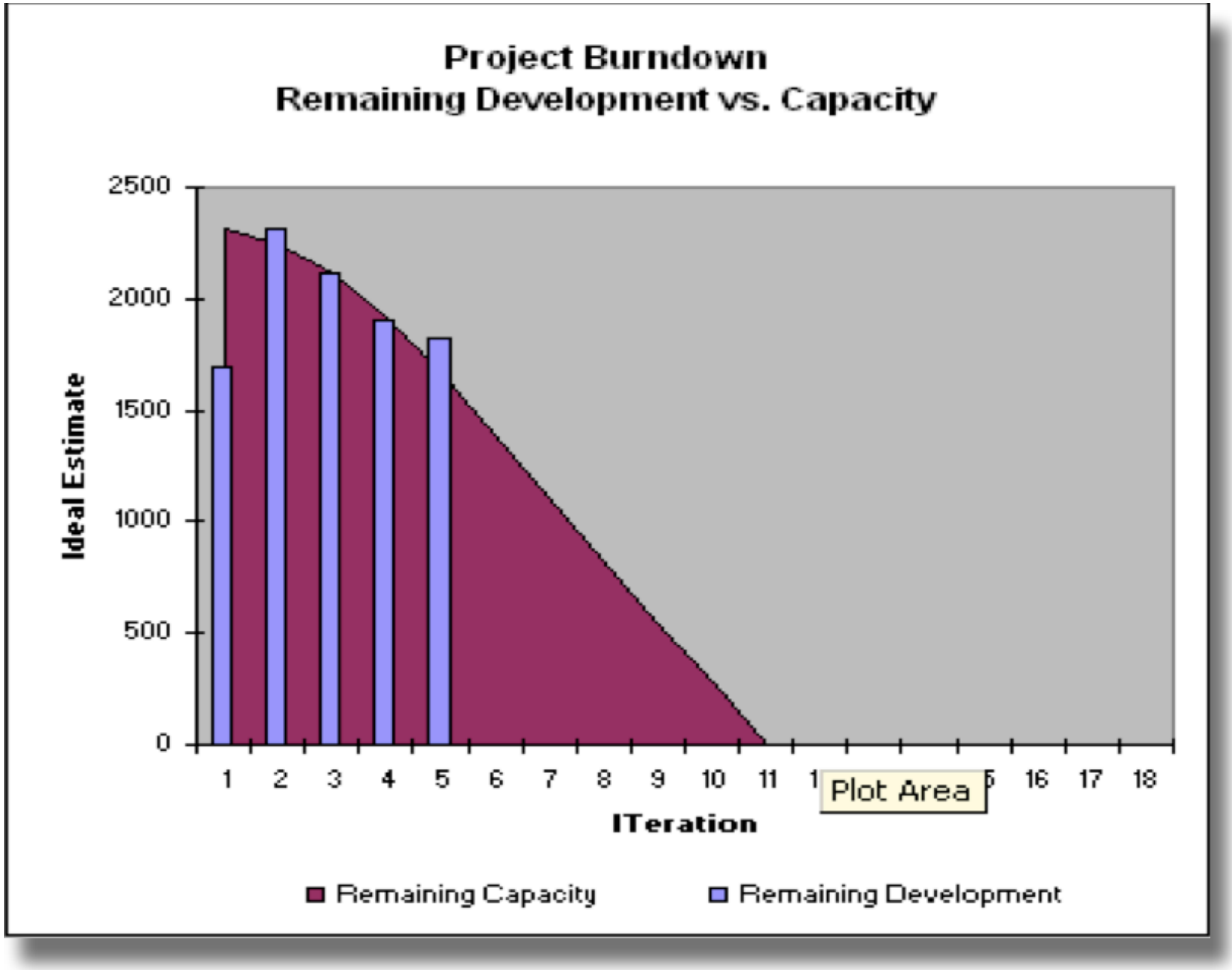
project-level metrics




Sparky’s
spreadsheet


Will the lines meet?
Total iteration estimates vs. Dev Complete
Number of Cards
3000
2500
2000
1500
1000
500
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
Iteration
Total Scope (Estimate)
Cards Dev Complete
Should be Dev Complete


Will the lines meet?
Total iteration estimates vs. BA Passed
Number of Cards
3000
2500
2000
1500
1000
500
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
Iteration
Total Scope (Estimate)
Cards passed by BA's
Should be passed (BA)


Project Burndown
Remaining Development vs. Capacity
Ideal Estimate
2500
2000
1500
1000
500
0
1
2
3
4
5
6
7
8
9
10
11
16
17
18
Plot Area
ITeration
Remaining Capacity
Remaining Development

Iteration that just ended

4

Capacity Remaining
(Ideal Hours)

1672

Do you have too much work left, for the days you have remaining?
(Graph represented on the bottom of this page)

Remaining Work in hours days, cards not Dev Complete
(sum ITKO estimates)

	Priority
256	A
1388	A+B
1828	A+B+C

Use for the Iteration Kick Off Meeting

Purpose: Plan out how many folks you have vs. how much work you have, per iteration
Developer team size and vacation days calculates from your budget sheet!
Note: This table determines your development capacity!
Changes to your expected velocity will change how much work you can expect to finish

Planning Iteration Work Load

Iteration	Development Start	Expected Load Factor	Team Sz. (Developers)	Developer Vacation Hours	Ideal Hours	*ITKO estimates for the cards assigned to this iteration*	Hangover Estimate (Unhide columns to include other Dev. time constraints)	Over/Under
1	Tuesday, July 05, 2005	0.30	3.0	0.0	72.0	72		-
2	Tuesday, July 19, 2005	0.40	4.0	0.0	128.0	128		-
3	Tuesday, August 02, 2005	0.50	5.0	0.0	200.0	192		(8.00)
4	Tuesday, August 16, 2005	0.50	6.0	0.0	240.0	264		24.00
5	Tuesday, August 30, 2005	0.50	7.0	0.0	280.0	280		-
6	Tuesday, September 13, 2005	0.50	7.0	0.0	280.0	296		16.00
7	Tuesday, September 27, 2005	0.50	7.0	0.0	280.0	288		8.00
8	Tuesday, October 11, 2005	0.50	7.0	0.0	280.0	248		(32.00)
9	Tuesday, October 25, 2005	0.50	7.0	16.0	272.0	300		28.00
10	Tuesday, November 08, 2005	0.50	7.0	0.0	280.0	440		160.00
11	Tuesday, November 22, 2005		0.0	0.0	-	0		-
12	Tuesday, December 06, 2005		0.0	0.0	-	0		-
13	Tuesday, December 20, 2005		0.0	0.0	-	0		-
14	Tuesday, January 03, 2006		0.0	0.0	-	0		-
15	Tuesday, January 17, 2006		0.0	0.0	-	0		-
16	Tuesday, January 31, 2006		0.0	0.0	-	0		-
17	Tuesday, February 14, 2006		0.0	0.0	-	0		-
18	Tuesday, February 28, 2006		0.0	0.0	-	0		-
Unscheduled						0		-


the customer is always available...
...or a worthwhile substitute

customer proxy

80% business, 20% technical

“feed” the development process

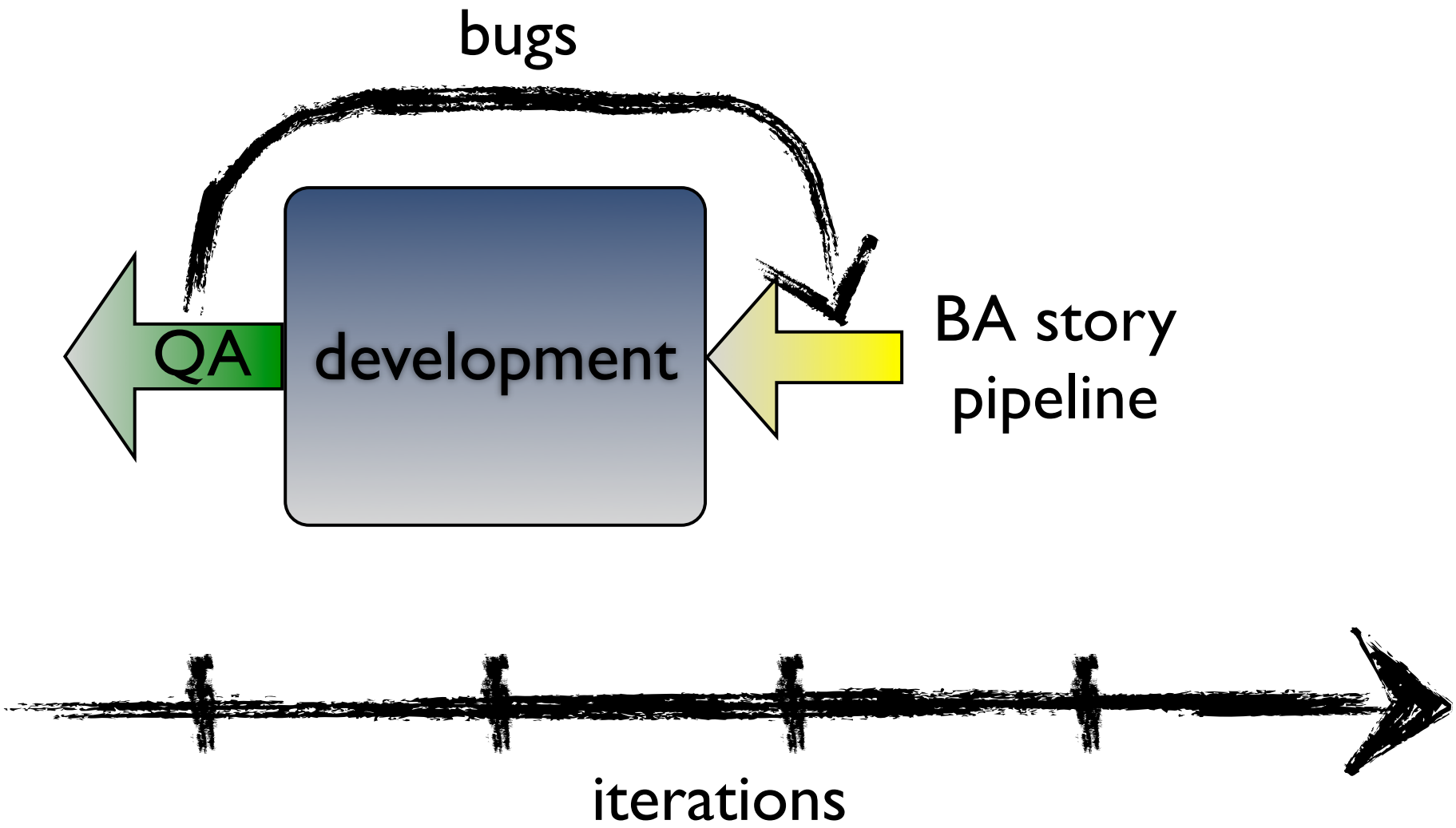
business analysts

(local) subject matter experts

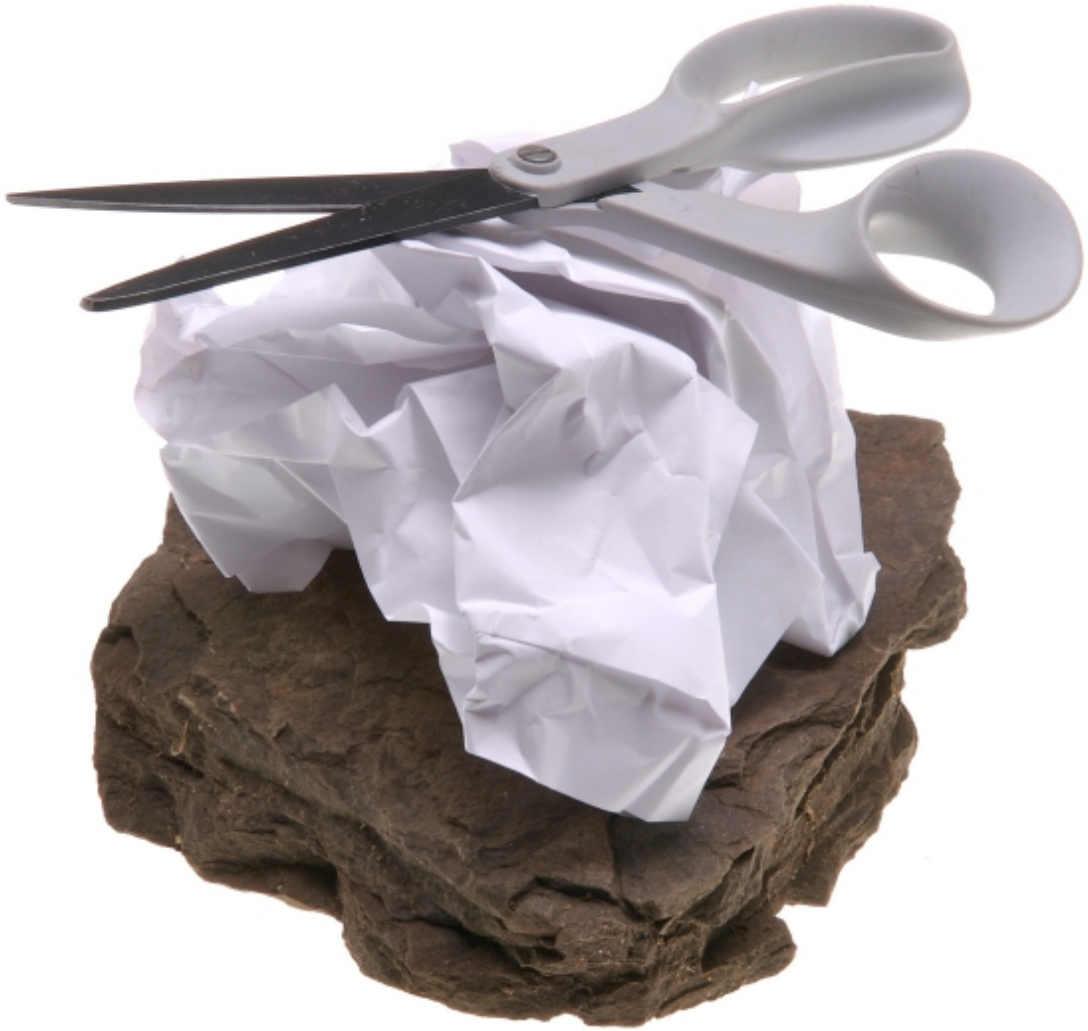
instant answer source

requirements gathering is a lossy compression algorithm

moving window



design practices



Rock is for Rookies: males have a tendency to lead with Rock on their opening throw.

boring

anticipatory design

fear

why is simplicity hard?

cleverness mixed with

irrational attachment

choose a system metaphor

domain driven design's *ubiquitous language*

class-responsibility-collaboration cards

alternative to UML

anything but UML!

use crc cards for design sessions

captures just what you need

deprecated by technology

design tools

what about...

documentation

useful

succinct

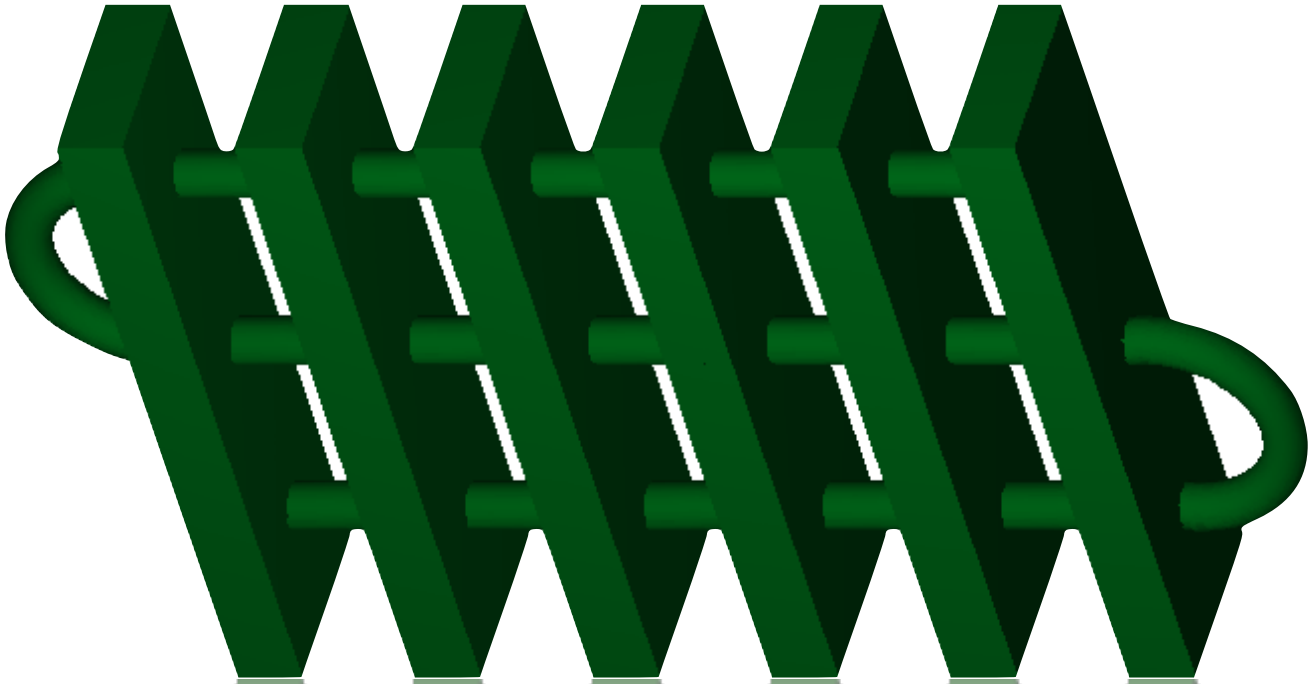
create spike solutions to reduce risk

not prototypes!

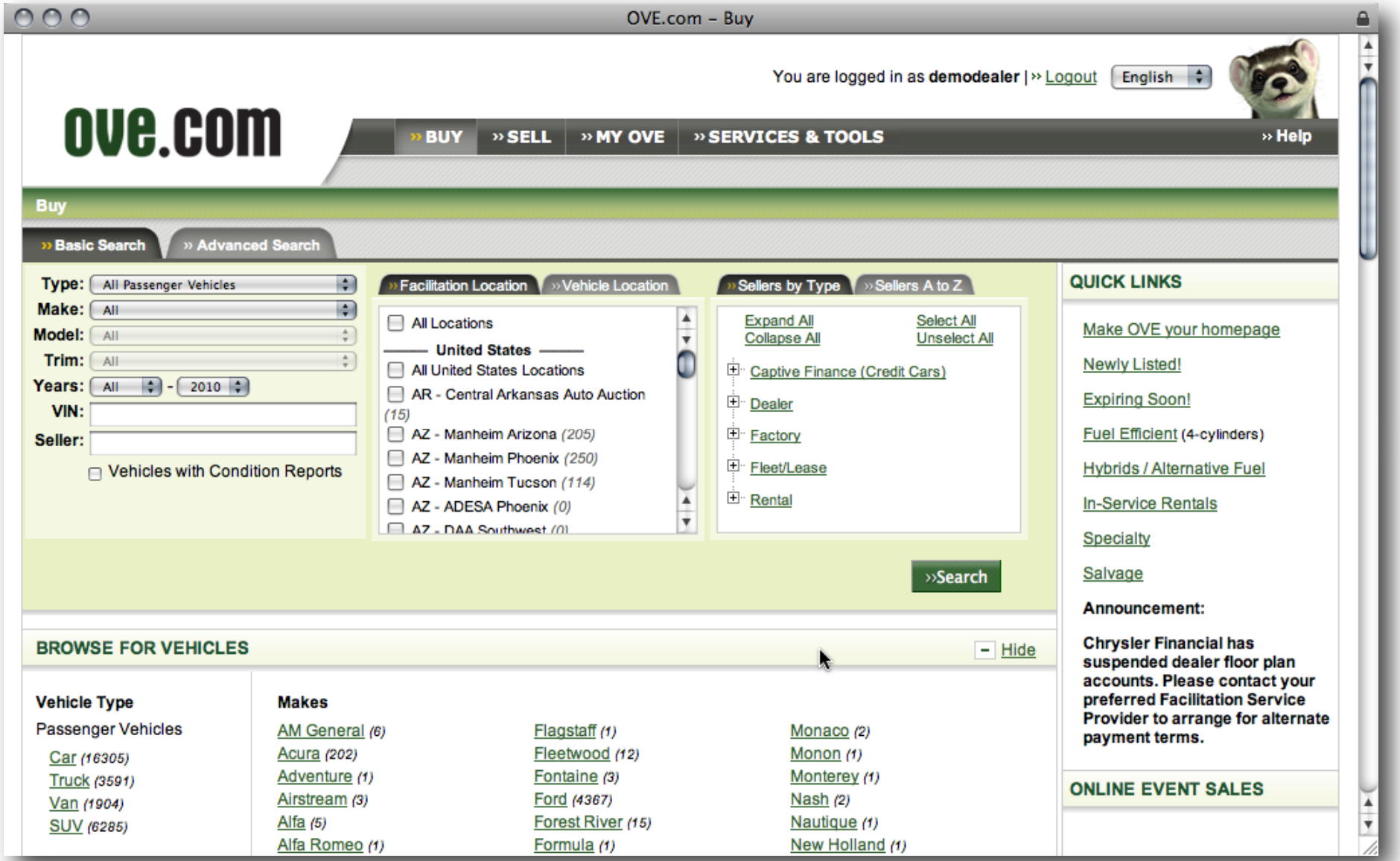
no functionality added early

yagni

don’t build frameworks



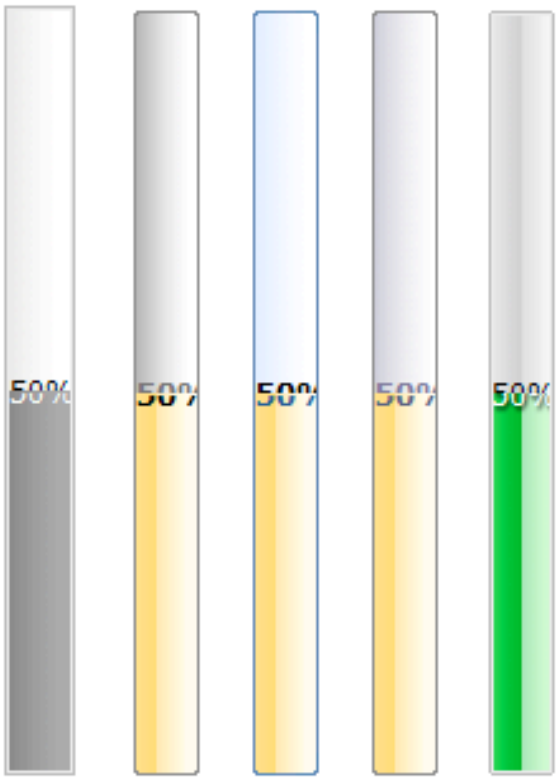
extract them

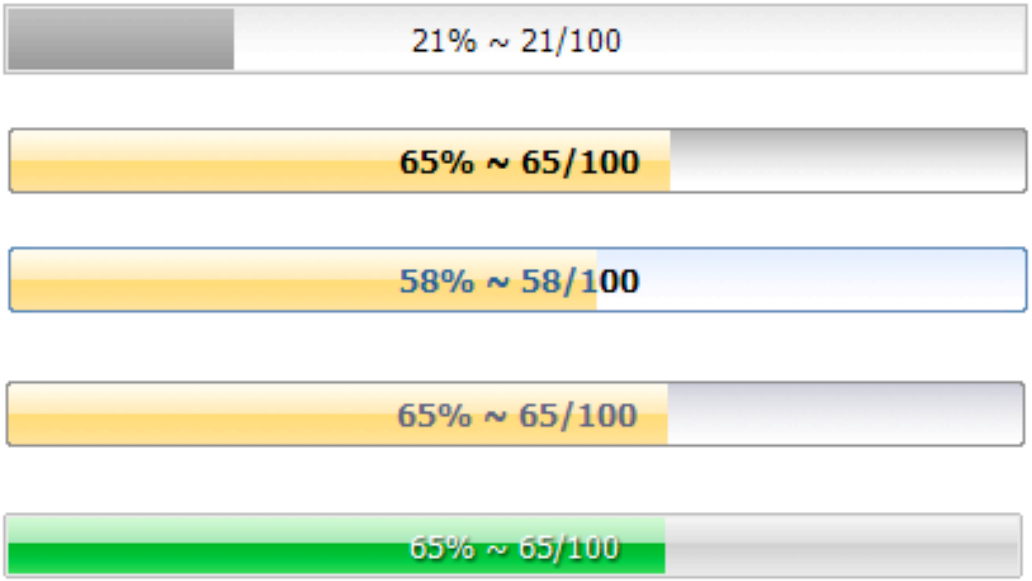

OVE.com – Buy
You are logged in as demodealer | » Logout
English
ove.com
» BUY
» SELL
» MY OVE
» SERVICES & TOOLS
» Help
Buy
» Basic Search
» Advanced Search
Type: All Passenger Vehicles
Make: All
Model: All
Trim: All
Years: All - 2010
VIN:
Seller:
Vehicles with Condition Reports
» Facilitation Location
» Vehicle Location
All Locations
United States
All United States Locations
AR - Central Arkansas Auto Auction (15)
AZ - Manheim Arizona (205)
AZ - Manheim Phoenix (250)
AZ - Manheim Tucson (114)
AZ - ADESA Phoenix (0)
» Sellers by Type
» Sellers A to Z
Expand All
Collapse All
Select All
Unselect All
Captive Finance (Credit Cars)
Dealer
Factory
Fleet/Lease
Rental
»Search
QUICK LINKS
Make OVE your homepage
Newly Listed!
Expiring Soon!
Fuel Efficient (4-cylinders)
Hybrids / Alternative Fuel
In-Service Rentals
Specialty
Salvage
Announcement:
Chrysler Financial has suspended dealer floor plan accounts. Please contact your preferred Facilitation Service Provider to arrange for alternate payment terms.
ONLINE EVENT SALES
BROWSE FOR VEHICLES
Hide
Vehicle Type
Passenger Vehicles
Car (16305)
Truck (3591)
Van (1904)
SUV (6285)
Makes
AM General (6)
Acura (202)
Adventure (1)
Airstream (3)
Alfa (5)
Alfa Romeo (1)
Flagstaff (1)
Fleetwood (12)
Fontaine (3)
Ford (4367)
Forest River (15)
Formula (1)
Monaco (2)
Monon (1)
Monterey (1)
Nash (2)
Nautique (1)
New Holland (1)

case study


evolution
of
asynchronous
messaging

progress bars & async upload

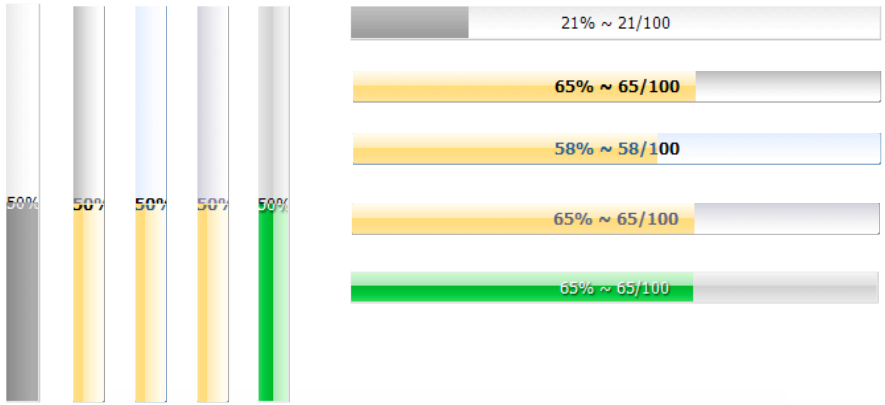




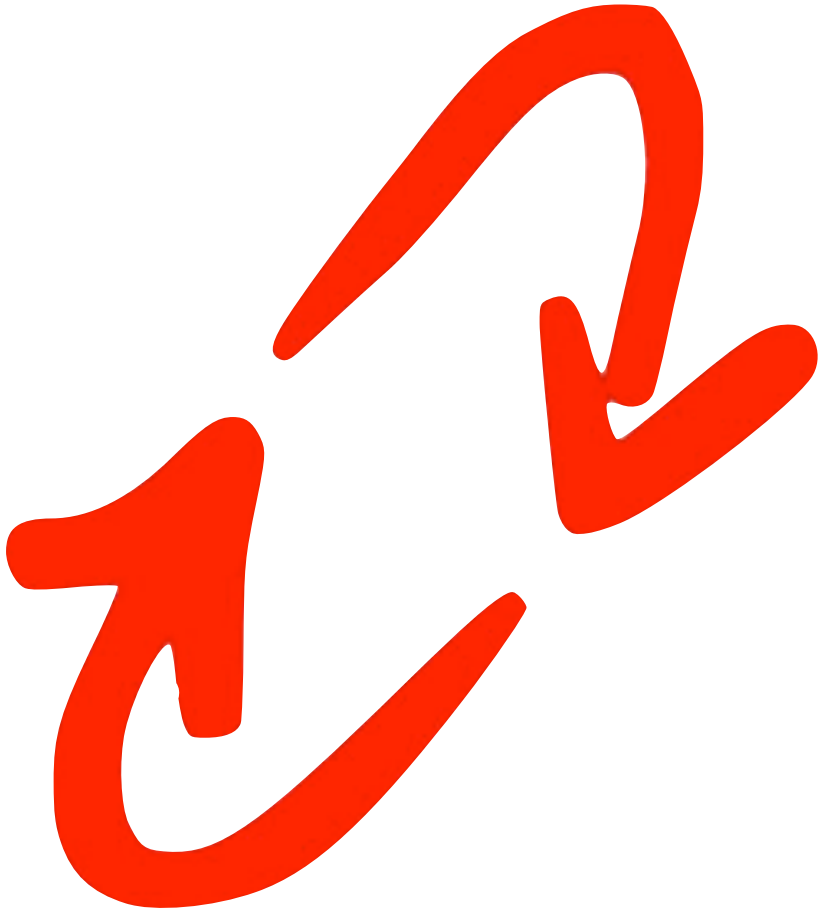
backgrounDrb

http://backgroundrb.rubyforge.org/

3 kinds


21% ~ 21/100
65% ~ 65/100
58% ~ 58/100
65% ~ 65/100
65% ~ 65/100

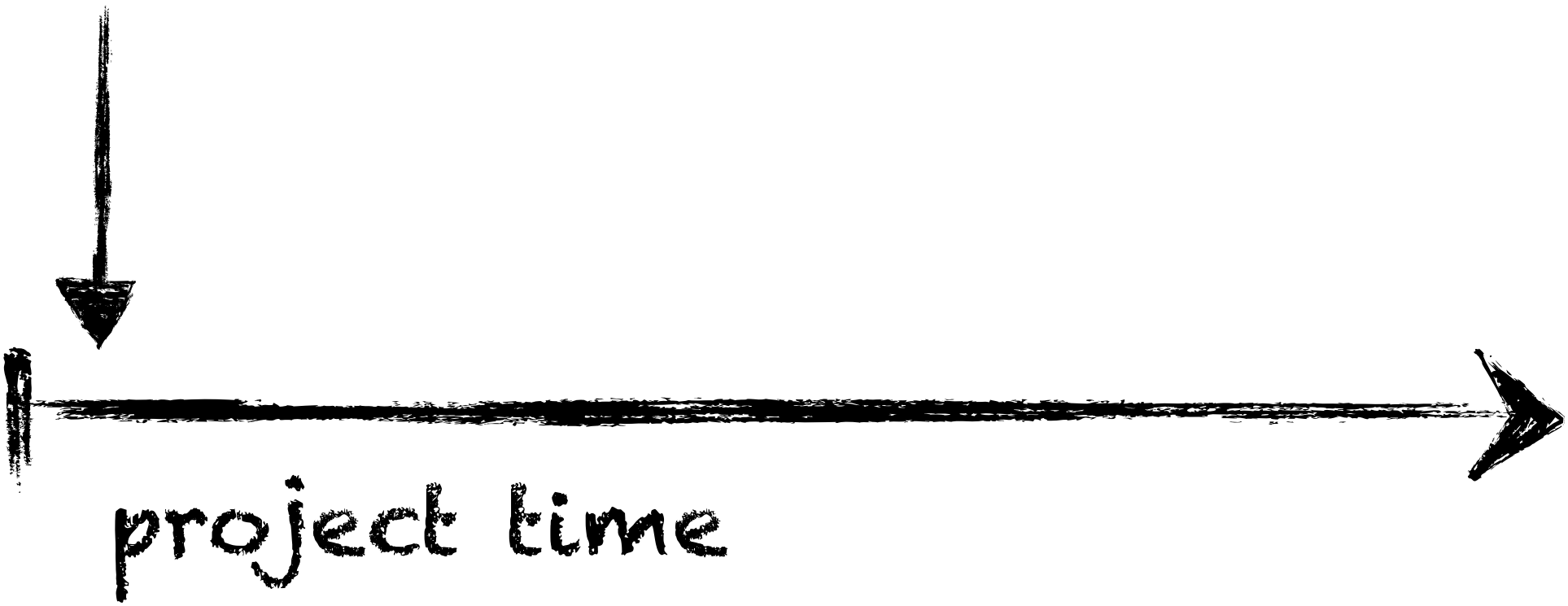

60
55
5
50
10
45
15
40
20
35
25
30

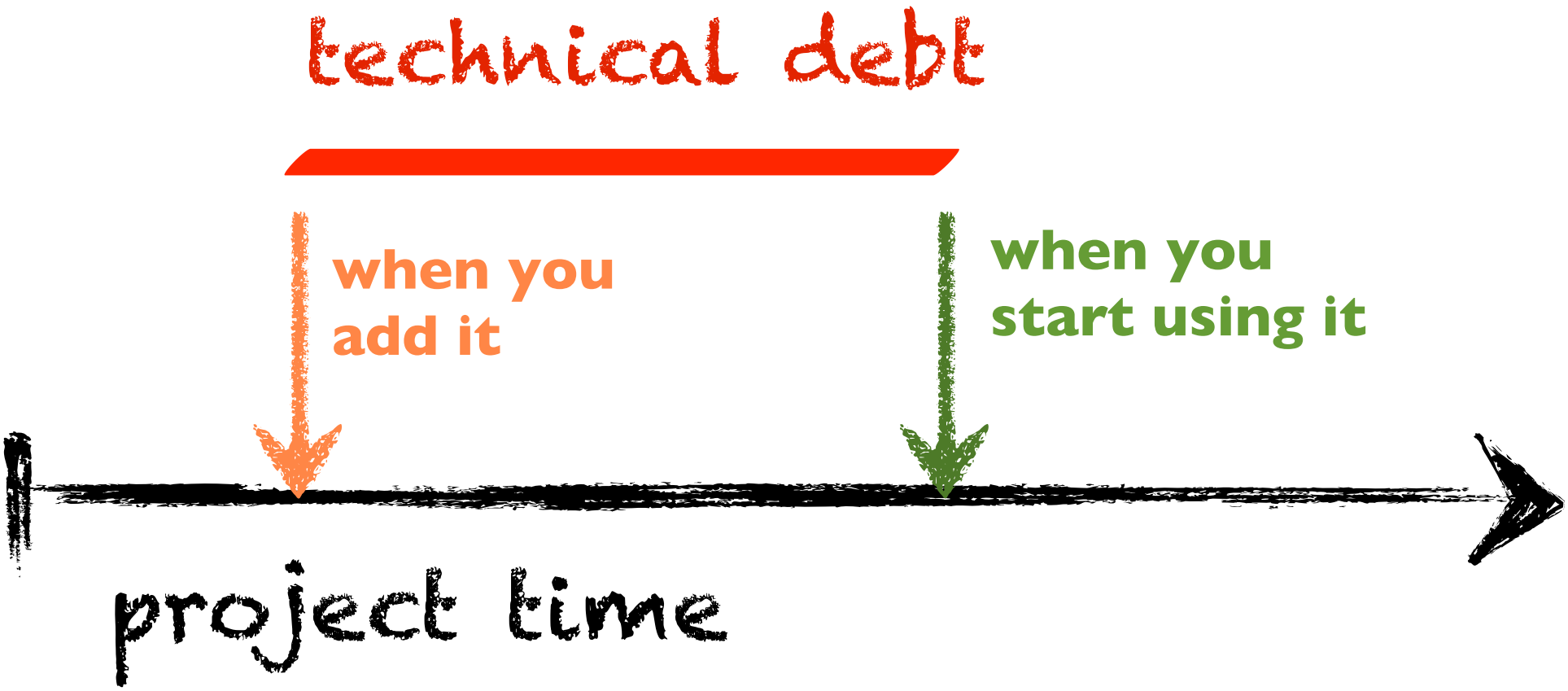



(Starling)
switch to a real
messaging queue

don’t know what we don’t know

“buy the fanciest one we can” (just in case)



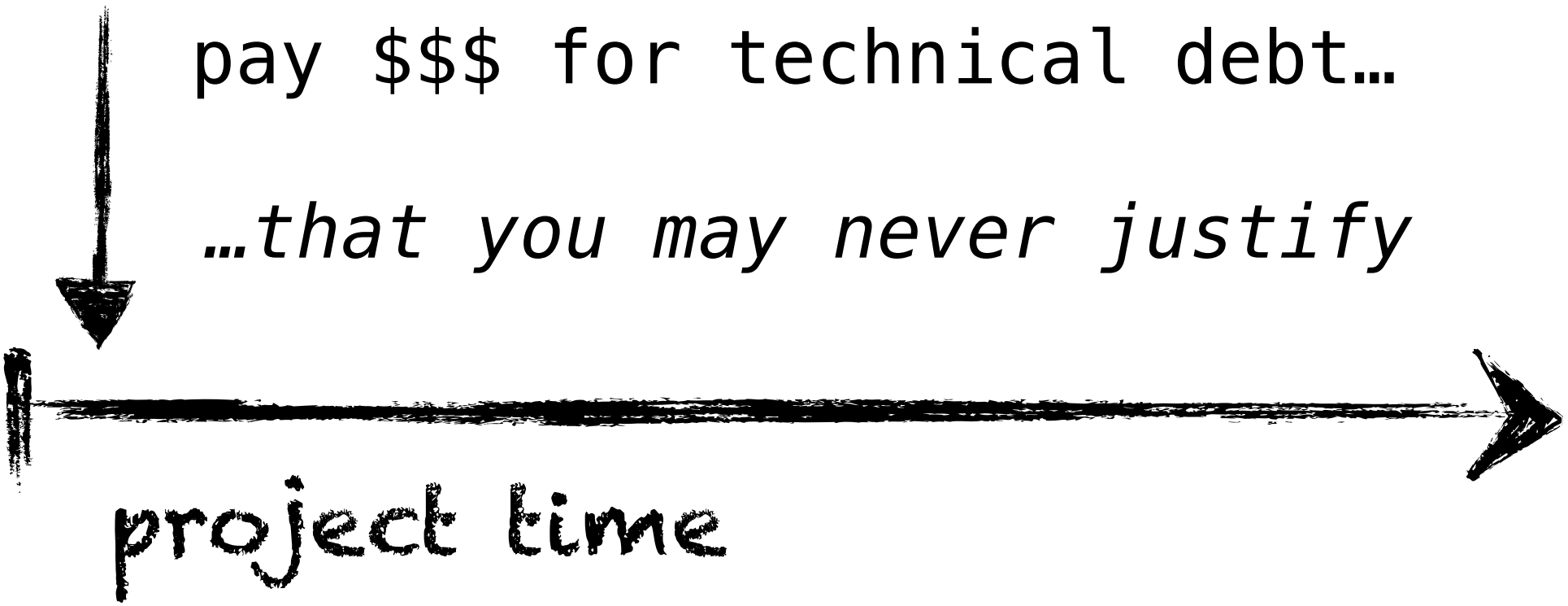

technical debt
when you add it
when you start using it
project time

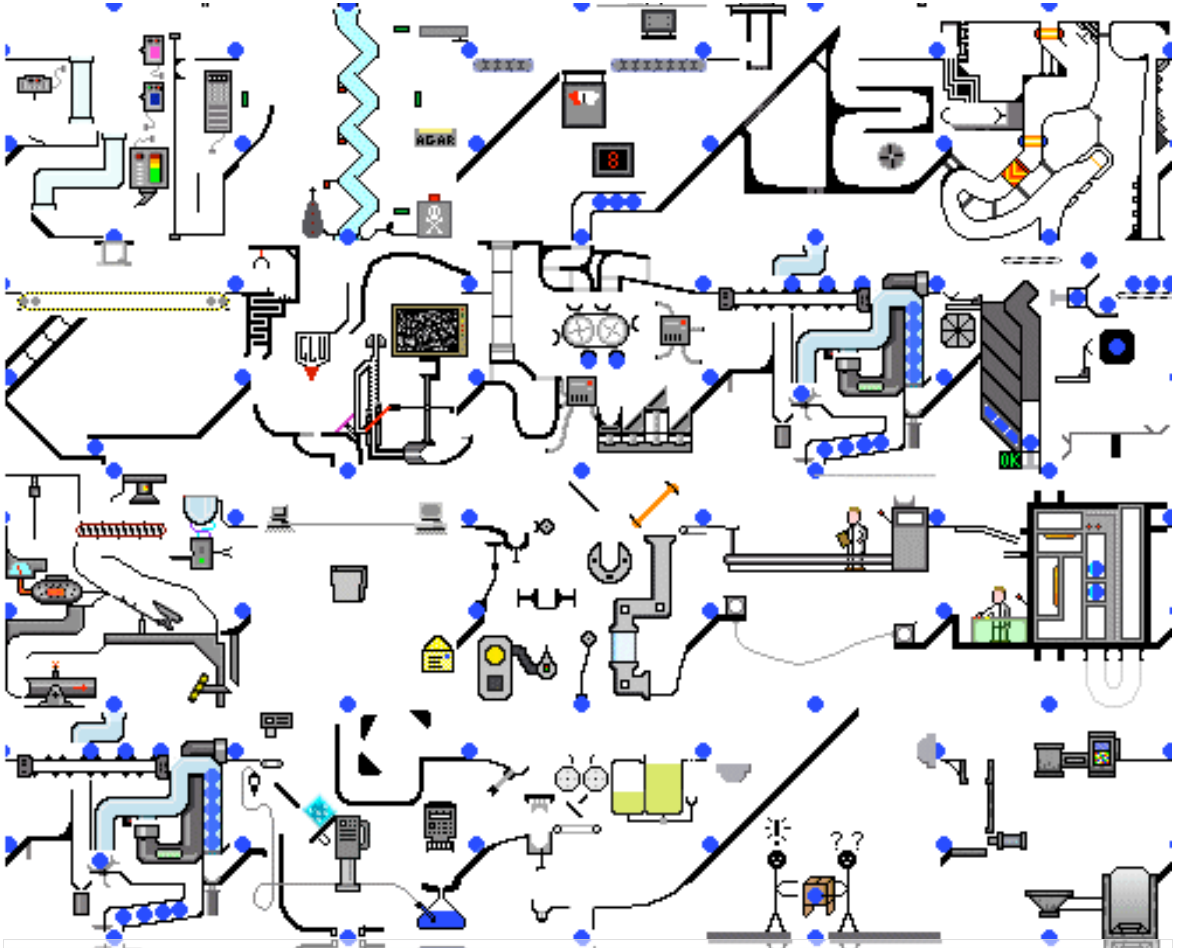
don’t know what we don’t know

“buy the fanciest one we can” (just in case)

pay $$$ for technical debt…

…that you may never justify



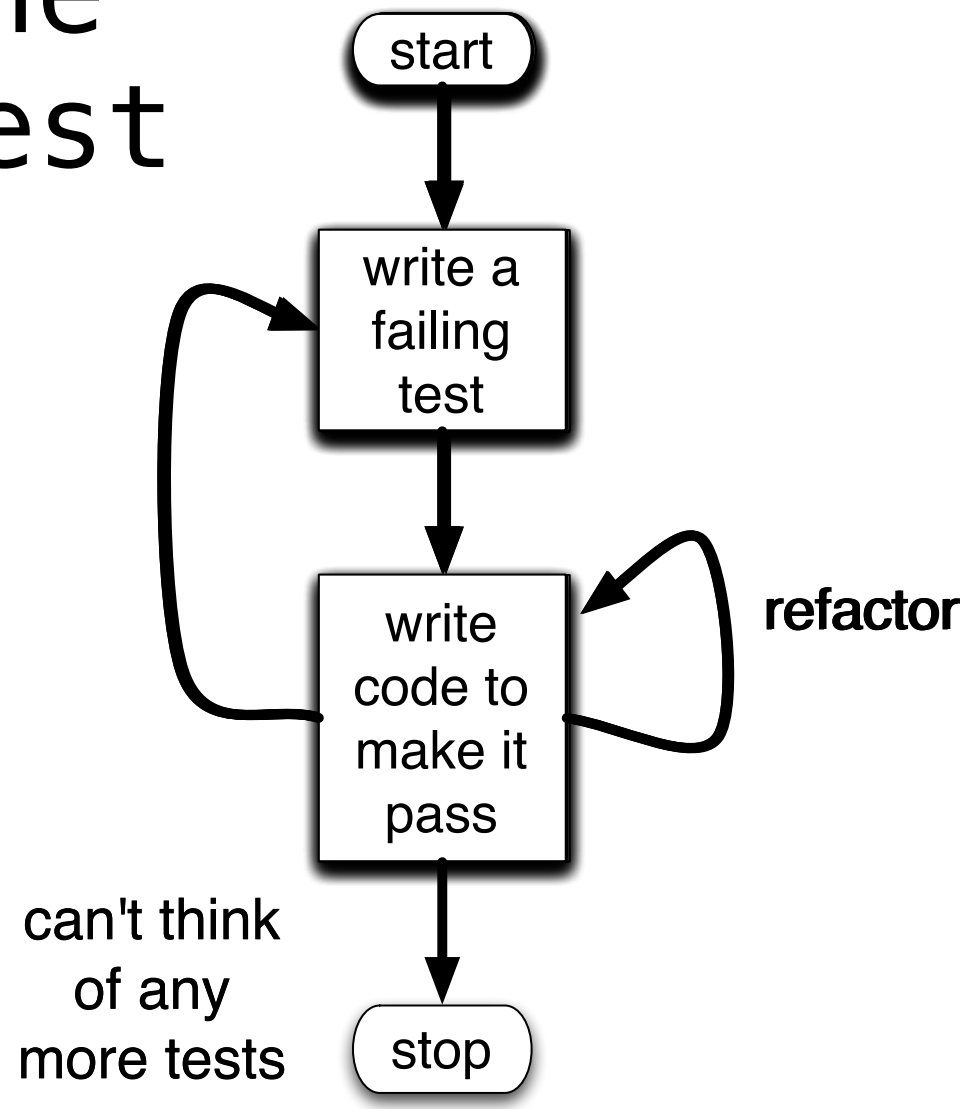


trying to predict the future leads to over-engineering

TDD, Design, & Velocity

Scissors on First: play scissors as your opening move against a more experienced player.

code the unit test first



red green refactor

test driven *design*

more about design than testing

design will emerge from tests

better abstractions

less accidental complexity

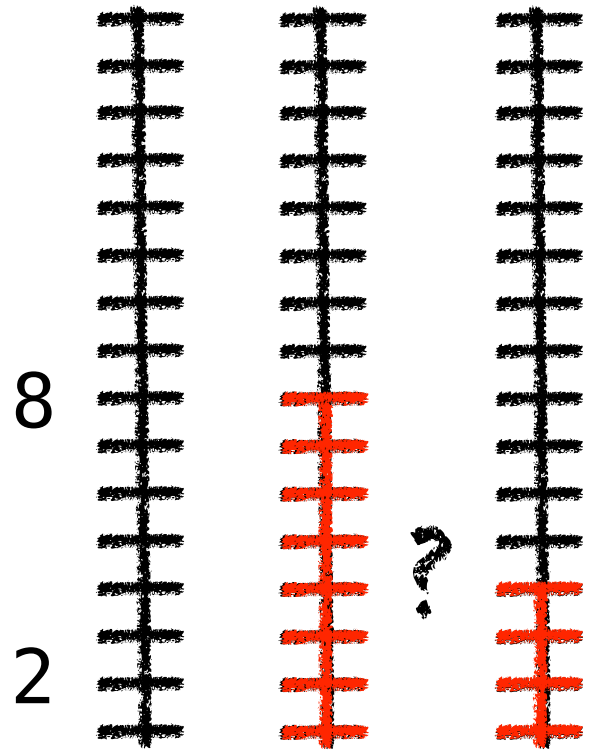
atomic understanding of intent

perfect number case study

∑ of the factors == number
(not including the number)

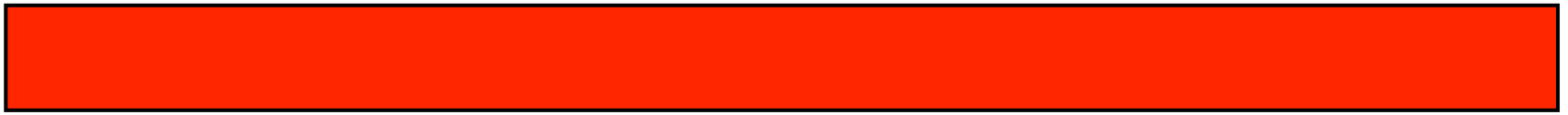
test-after, 1st pass

```
public class PerfectNumberFinder1 {
    public static boolean isPerfect(int number) {
        // get factors
        List<Integer> factors = new ArrayList<Integer>();
        factors.add(1);
        factors.add(number);
        for (int i = 2; i < number; i++)
            if (number % i == 0)
                factors.add(i);

        // sum factors
        int sum = 0;
        for (int n : factors)
            sum += n;

        // decide if it's perfect
        return sum - number == number;
    }
}
```



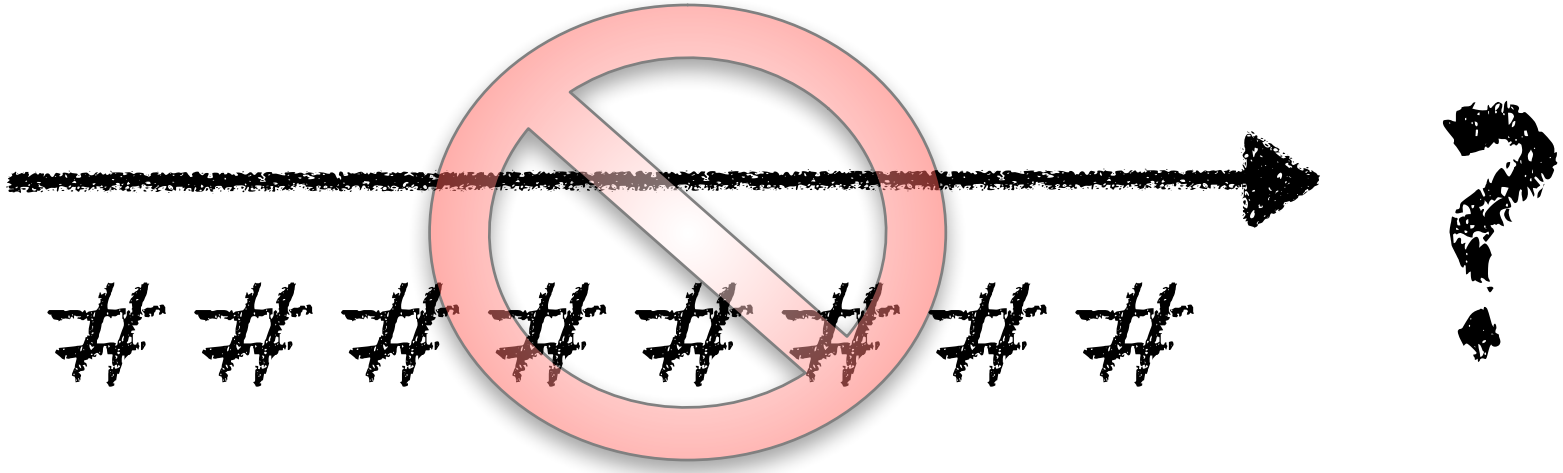
```
public class PerfectNumberFinder2 {
    public static boolean isPerfect(int number) {
        // get factors
        List<Integer> factors = new ArrayList<Integer>();
        factors.add(1);
        factors.add(number);
        for (int i = 2; i <= sqrt(number); i++)
            if (number % i == 0) {
                factors.add(i);
                factors.add(number / i);
            }

        // sum factors
        int sum = 0;
        for (int n : factors)
            sum += n;

        // decide if it's perfect
        return sum - number == number;
    }
}
```

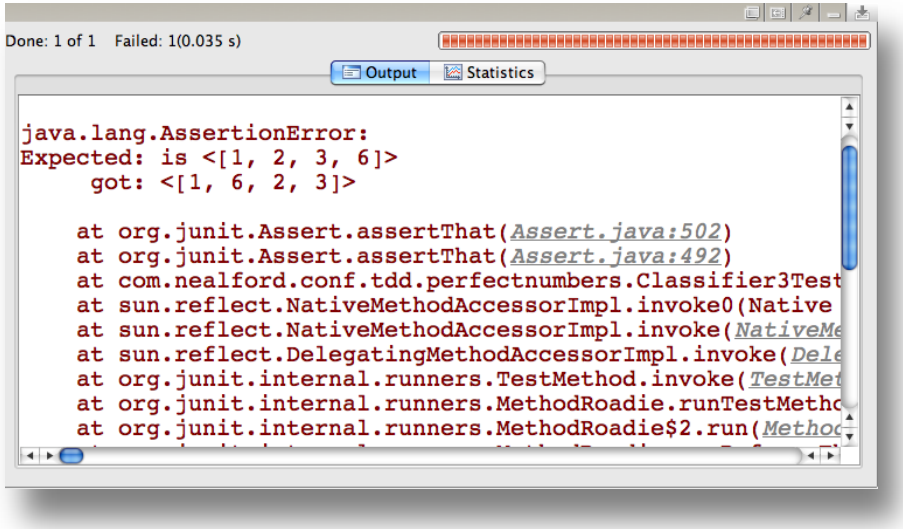
whole-number square roots

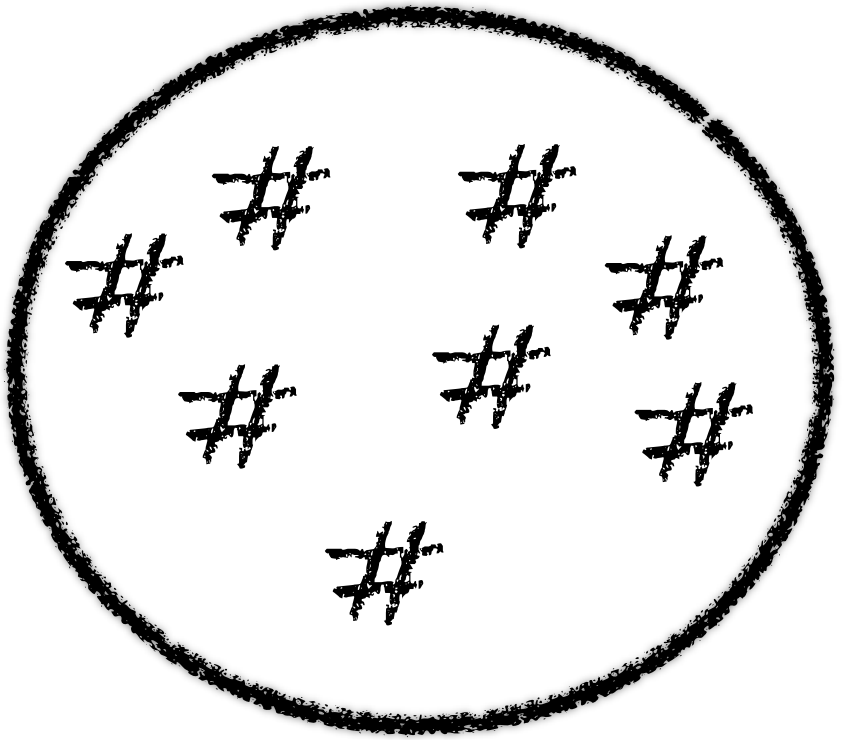


```
public class PerfectNumberFinder2 {
    public static boolean isPerfect(int number) {
        // get factors
        List<Integer> factors = new ArrayList<Integer>();
        factors.add(1);
        factors.add(number);
        for (int i = 2; i <= sqrt(number); i++)
            if (number % i == 0) {
                factors.add(i);
                // guard against whole-number square roots
                if (number / i !=  i)
                    factors.add(number / i);
            }

        // sum factors
        int sum = 0;
        for (int n : factors)
            sum += n;

        // decide if it's perfect
        return sum - number == number;
    }
}
```

```
public class Classifier6 {
    private Set<Integer> _factors;
    private int _number;

    public Classifier6(int number) {
        if (number < 1)
            throw new InvalidNumberException(
            "Can't classify negative numbers");
        _number = number;
        _factors = new HashSet<Integer>();
        _factors.add(1);
        _factors.add(_number);
    }

    private boolean isFactor(int factor) {
        return _number % factor == 0;
    }

    public Set<Integer> getFactors() {
        return _factors;
    }

    private void calculateFactors() {
        for (int i = 2; i < sqrt(_number) + 1; i++)
            if (isFactor(i))
                addFactor(i);
    }

    private void addFactor(int factor) {
        _factors.add(factor);
        _factors.add(_number / factor);
    }

    private int sumOfFactors() {
        calculateFactors();
        int sum = 0;
        for (int i : _factors)
            sum += i;
        return sum;
    }

    public boolean isPerfect() {
        return sumOfFactors() - _number == _number;
    }
```




Done: 1 of 1 Failed: 1(0.035 s)
Output
Statistics
java.lang.AssertionError:
Expected: is <[1, 2, 3, 6]>
got: <[1, 6, 2, 3]>
at org.junit.Assert.assertThat(Assert.java:502)
at org.junit.Assert.assertThat(Assert.java:492)
at com.nealford.conf.tdd.perfectnumbers.Classifier3Test
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMe
at sun.reflect.DelegatingMethodAccessorImpl.invoke(Dele
at org.junit.internal.runners.TestMethod.invoke(TestMet
at org.junit.internal.runners.MethodRoadie.runTestMetho
at org.junit.internal.runners.MethodRoadie$2.run(Method



test-after

```
for (int i = 2; i <= sqrt(number); i++)
    if (number % i == 0) {
        factors.add(i);
        // account for whole-number square roots
        if (number / i !=  i)
            factors.add(number / i);
    }
```

TDD

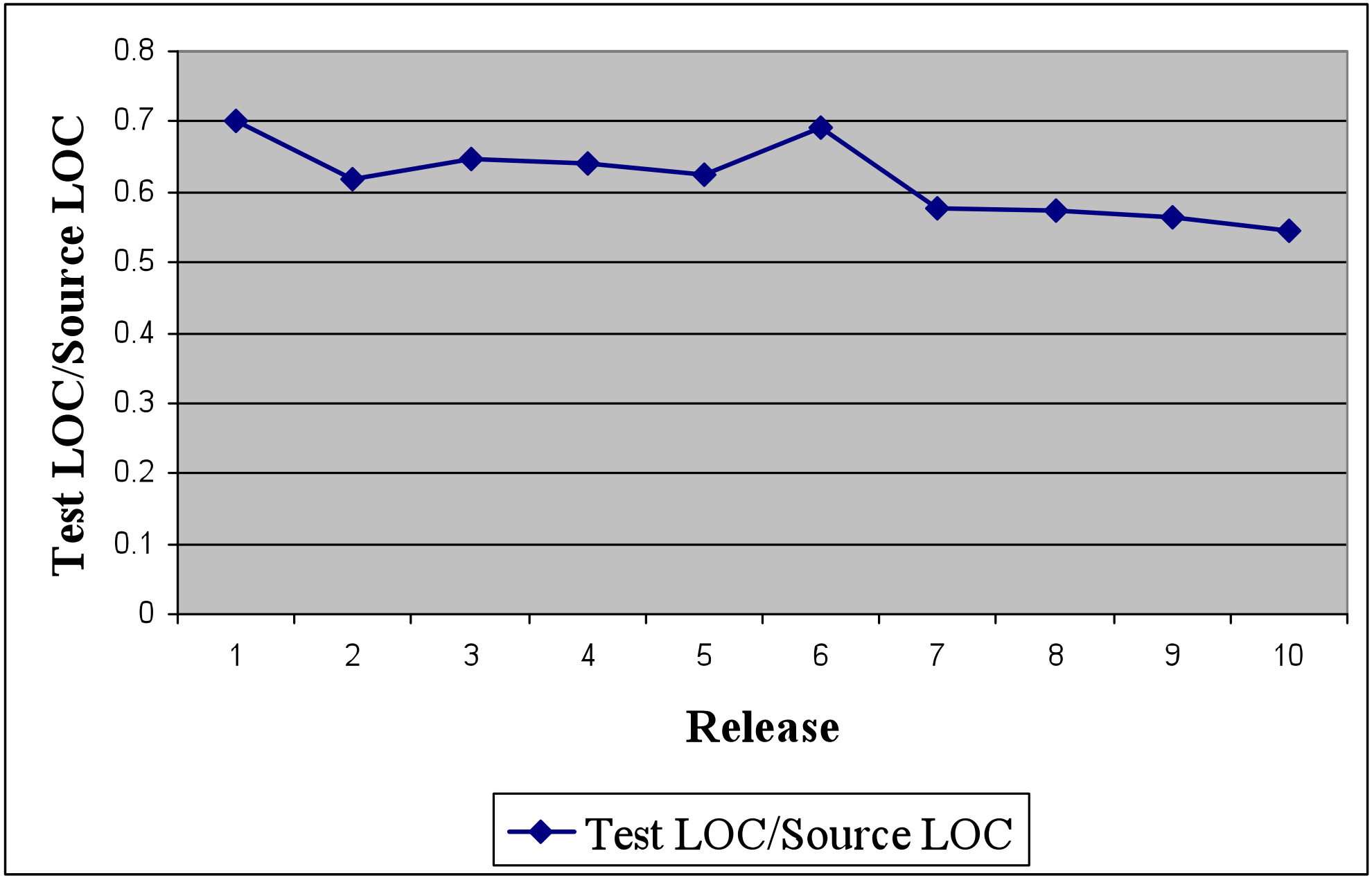
```
private void calculateFactors() {
    for (int i = 2; i < sqrt(_number) + 1; i++)
        if (isFactor(i))
            addFactor(i);
}

private void addFactor(int factor) {
    _factors.add(factor);
    _factors.add(_number / factor);
}
```

case studies

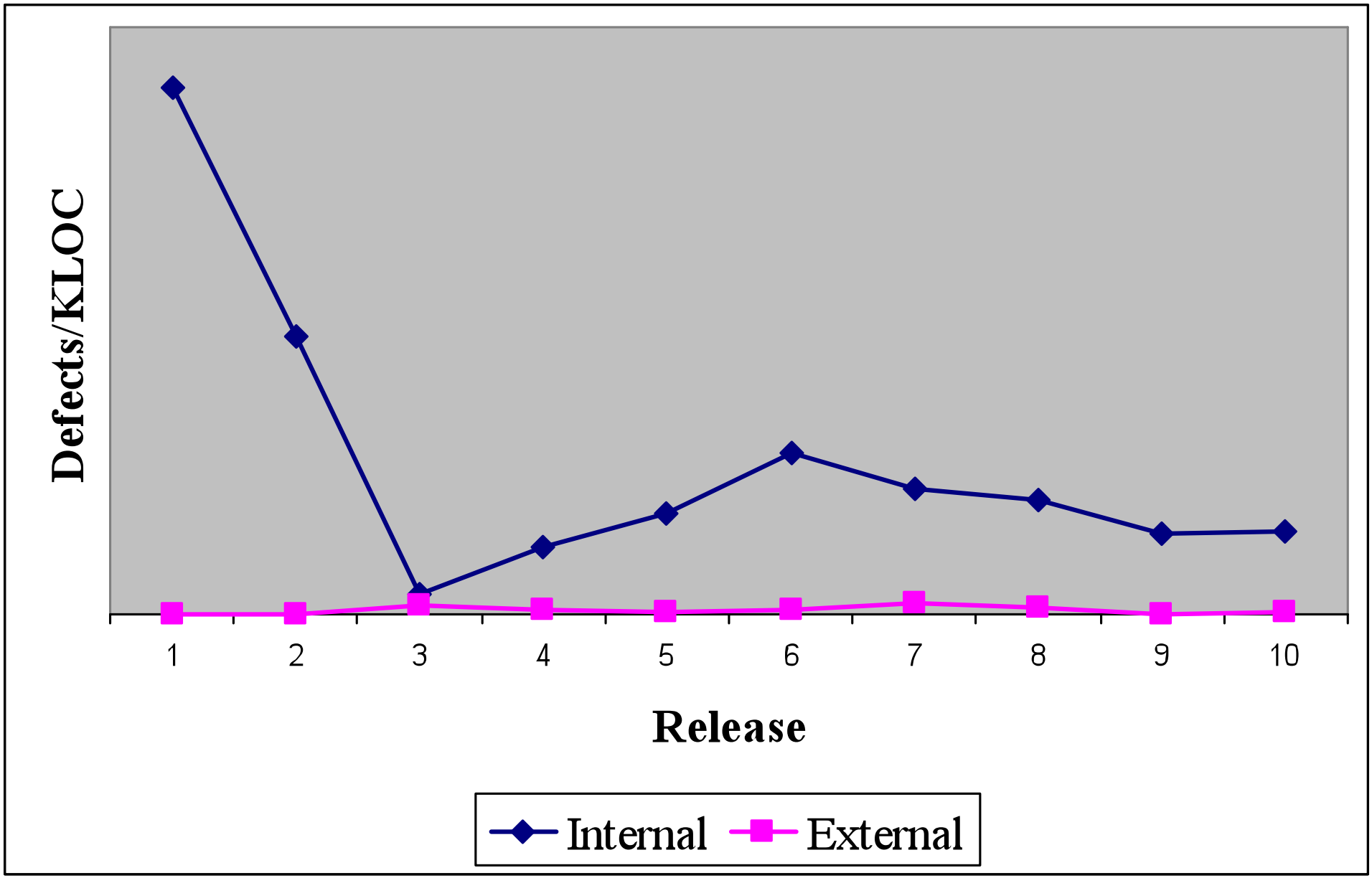
Dr. Laurie Williams

Associate Professor
North Carolina State University
Department of Computer Science

http://collaboration.csc.ncsu.edu/laurie/publications.html

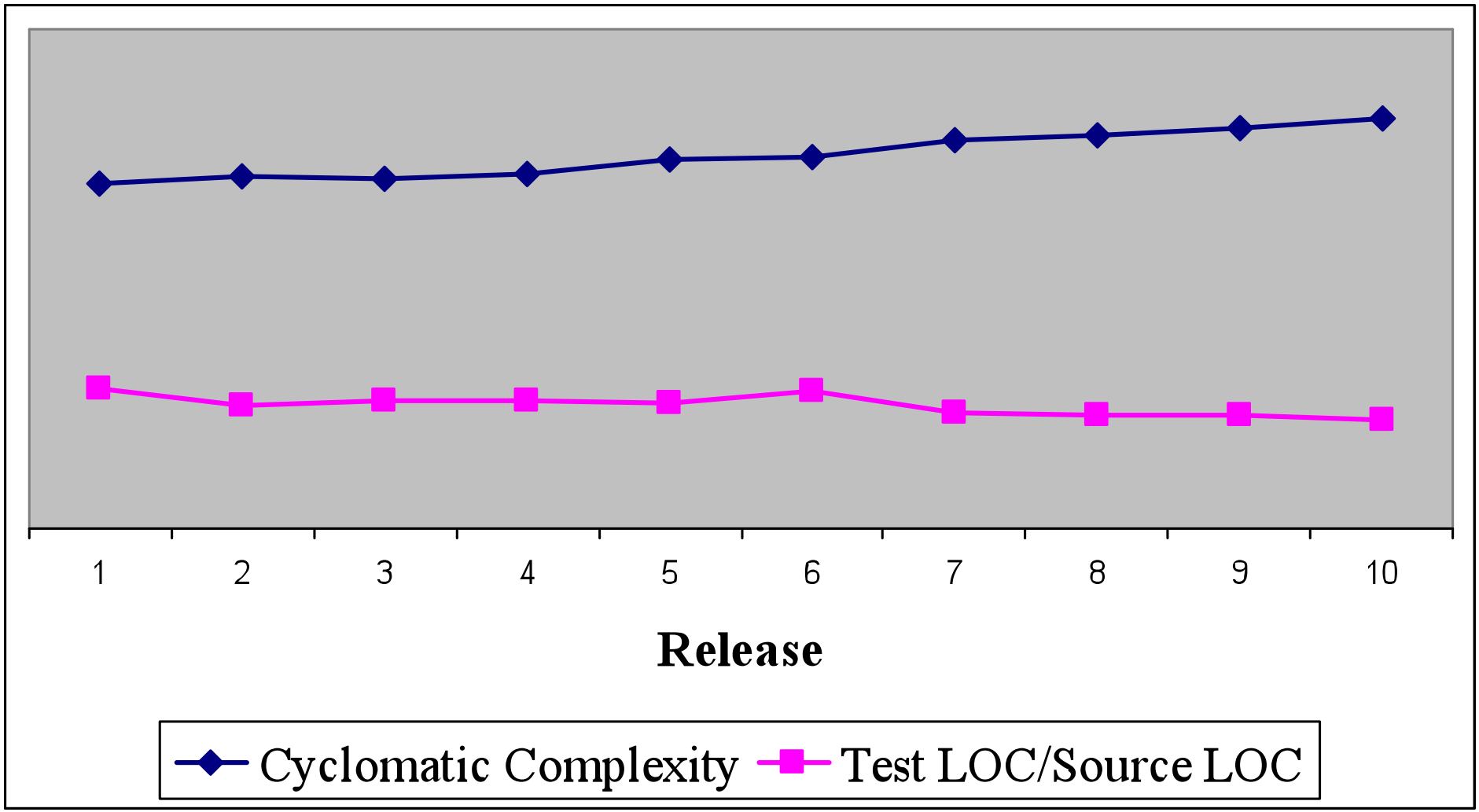


source: http://agile-carolinas.pbworks.com/f/WilliamsTDD.ppt



source: http://agile-carolinas.pbworks.com/f/WilliamsTDD.ppt

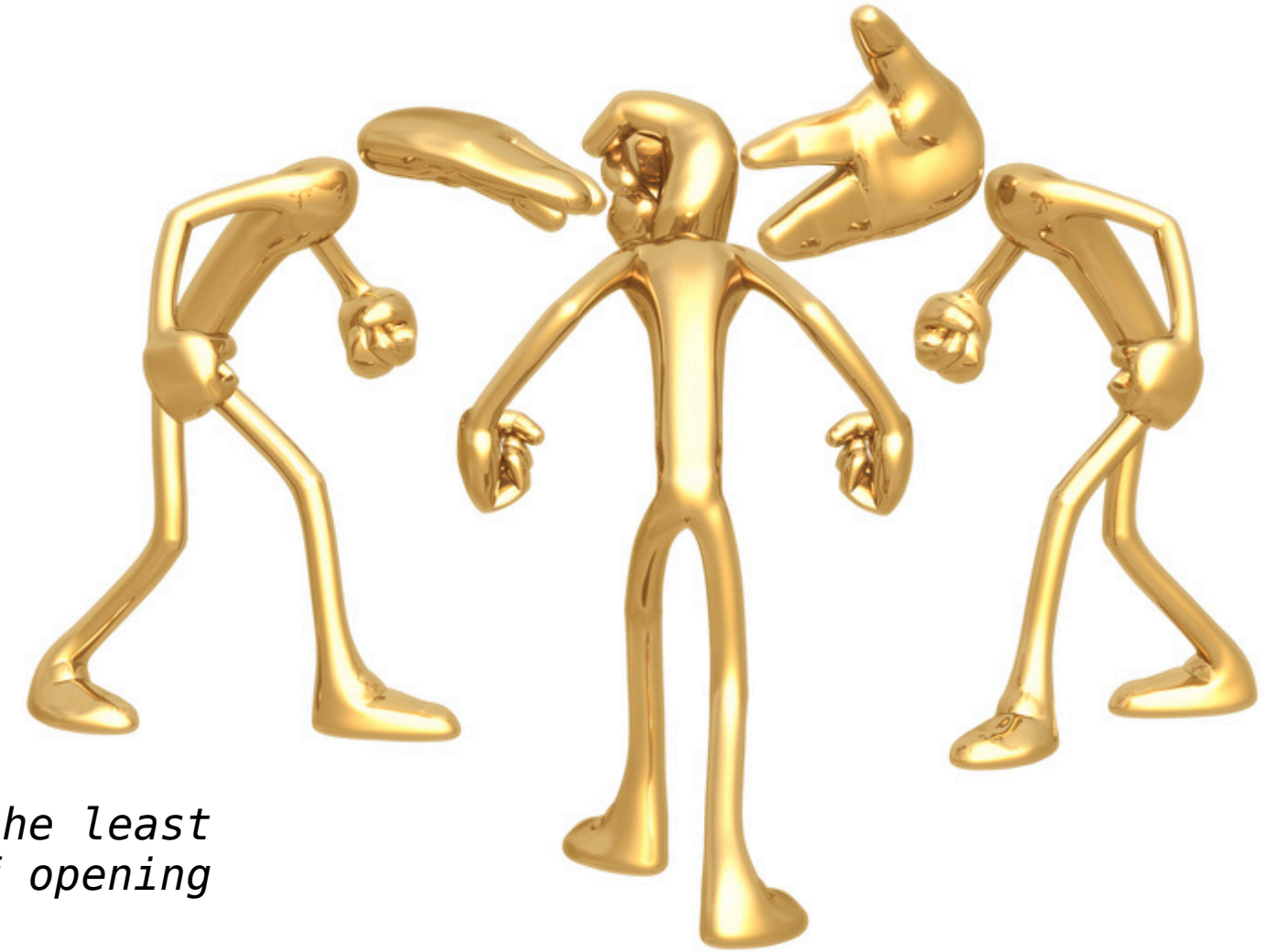
new “anti-aging” formula



source: http://agile-carolinas.pbworks.com/f/WilliamsTDD.ppt

writing more code allows you to go faster

pair programming mechanics



Paper is the least obvious of opening moves.

2 monitors

1 computer

2 mice

2 keyboards

pairing stations

not someone’s computer

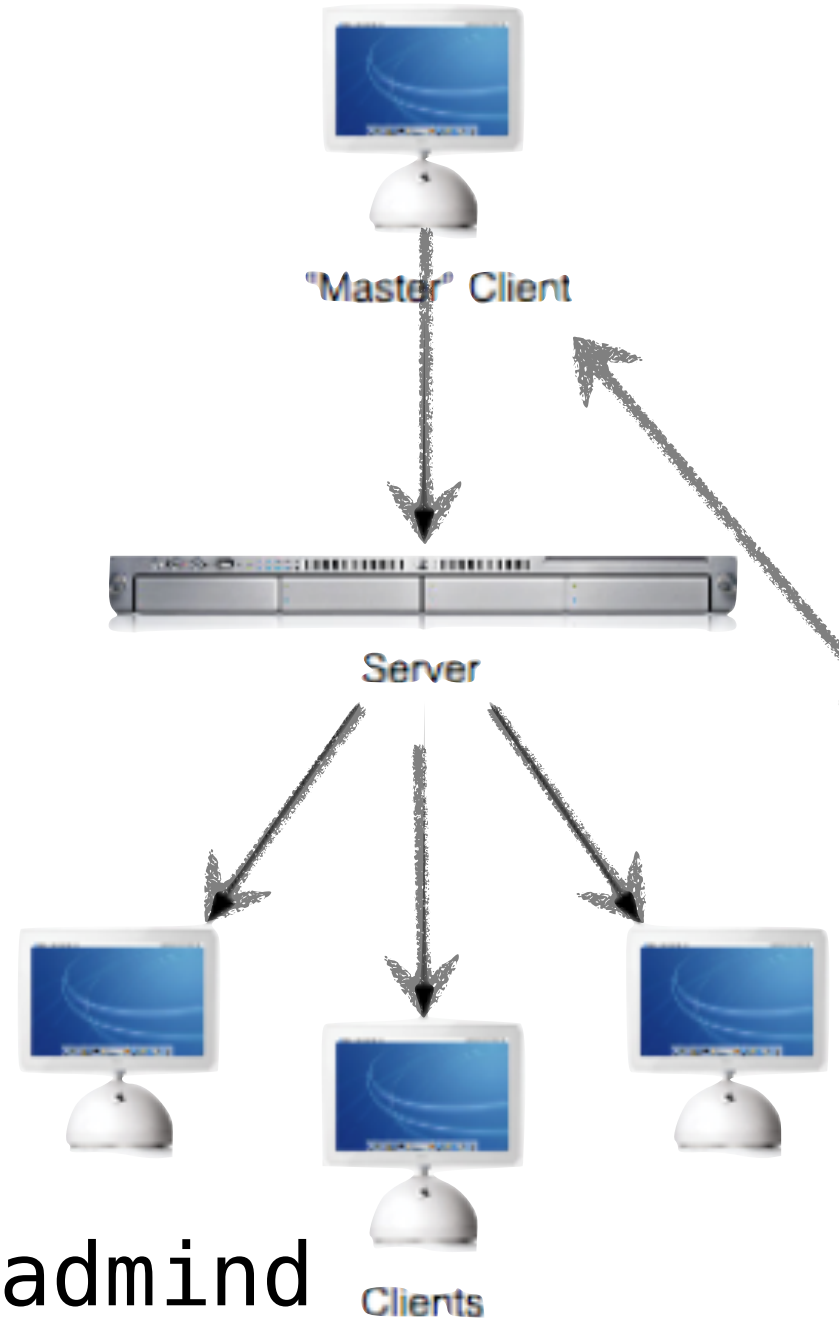
all the tools for development...

...and nothing else

pairing station ≠ your laptop*

mirrored...

mirrored workstations



automation

radmind

http://rsug.itd.umich.edu/software/radmind/

mozilla LABS Prism


driver
B
N

navigator

logistics

driver types & **narrates**

navigator thinks & interjects

design discussions in situ

no discussion > 10 mins w/o code

swap roles frequently

pair rotation

twice a day ⇔ every other day

tech lead picks effective pairs

reduces truck number metric

spreads knowledge across team

1 person must stay with story

you can only stay once/rotation

context update for the new pair

today’s new pair is tomorrow’s context keeper

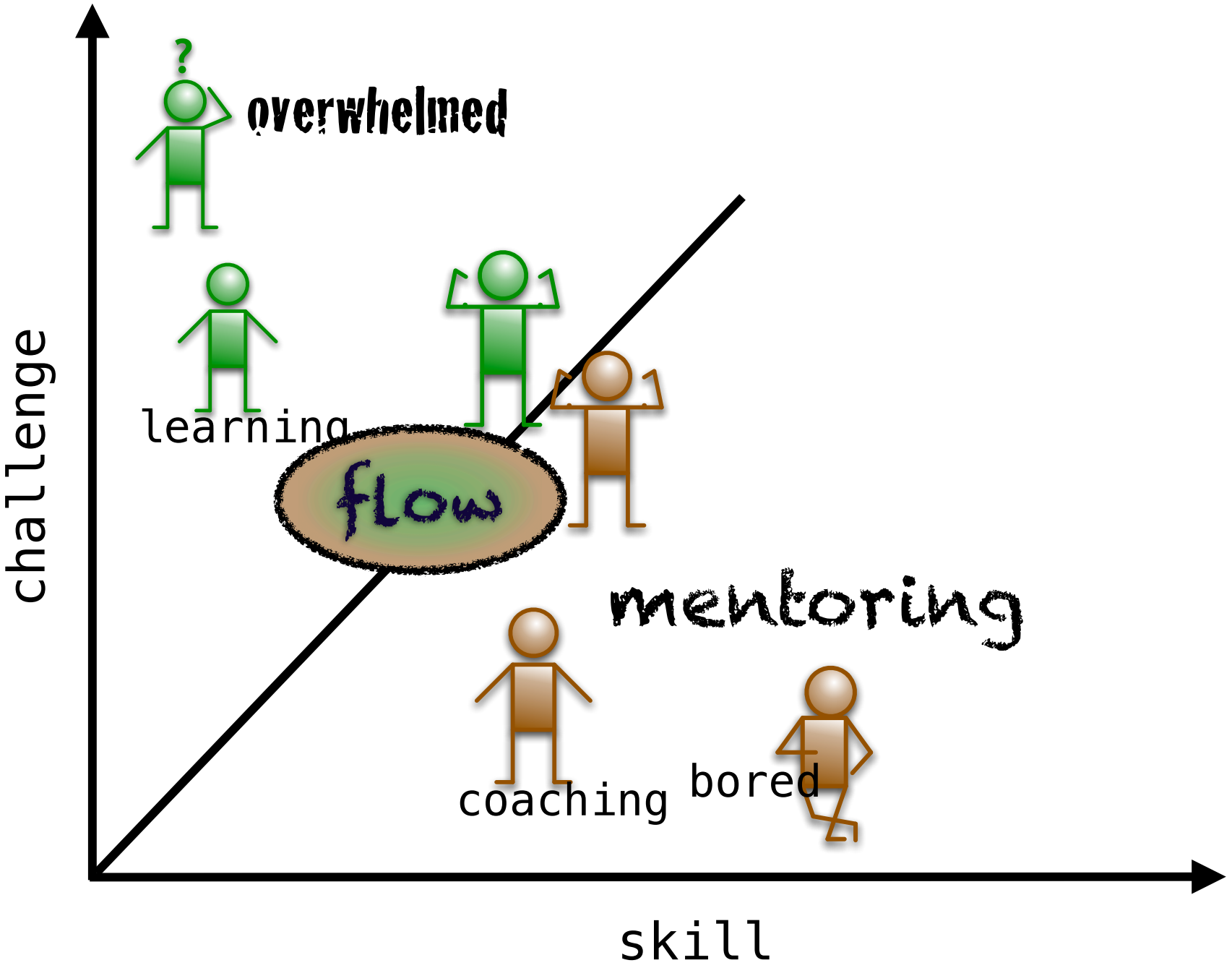
promiscuous knowledge


not!

what it's not!

2 people huddled over 1 computer

mentoring


?
overwhelmed
learning
flow
mentoring
coaching
bored
challenge
skill

what it's not!

2 people huddled over 1 computer

mentoring

keyboard domination

ping-pong pairing

what it's not!

2 people huddled over 1 computer

mentoring

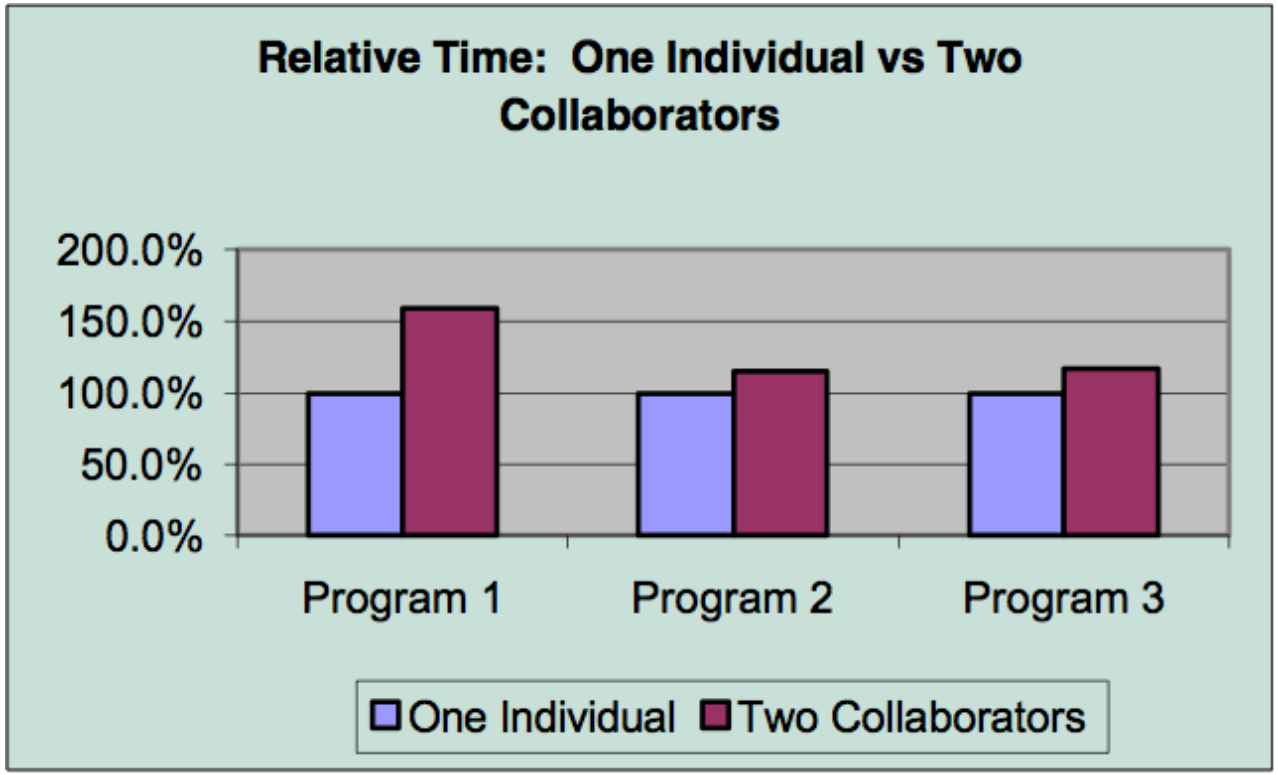
keyboard domination

pair marriages

> 10 mins of debate
  with no code

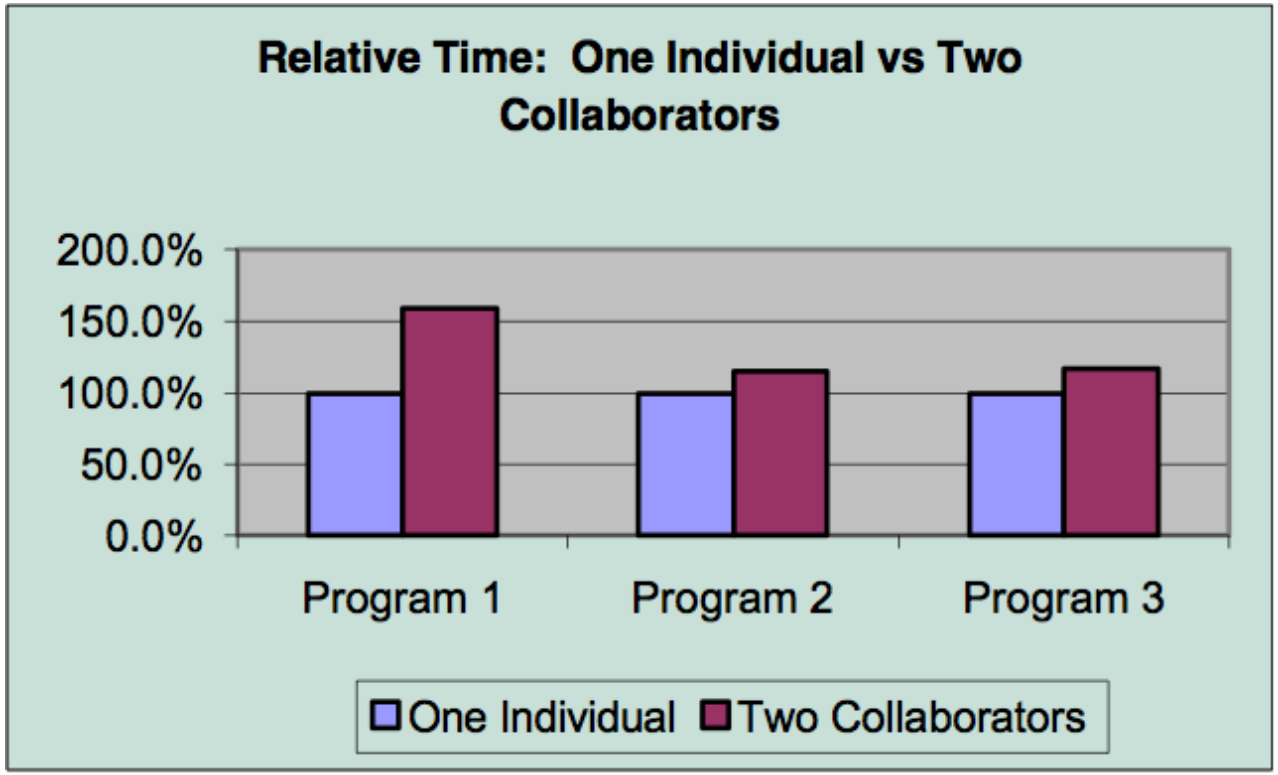
less productive

pair programming studies



after adjusting, pairs produced code 15% more slowly than individuals...

pair programming studies



...with 15% fewer defects

Williams et al

pairs 15% slower

15% fewer bugs

http://www.economist.com/displayStory.cfm?Story_ID=779429

"error free" code 70–85%

50% decrease in errors (30%–15%)

testing & debugging many times more $$$

more studies

Lui 2006 http://www.cs.utexas.edu/users/mckinley/305j/pair-hcs-2006.pdf

rigorous scientific experiment

novice-novice vs. novice solos
vs.
expert-expert vs. expert solo

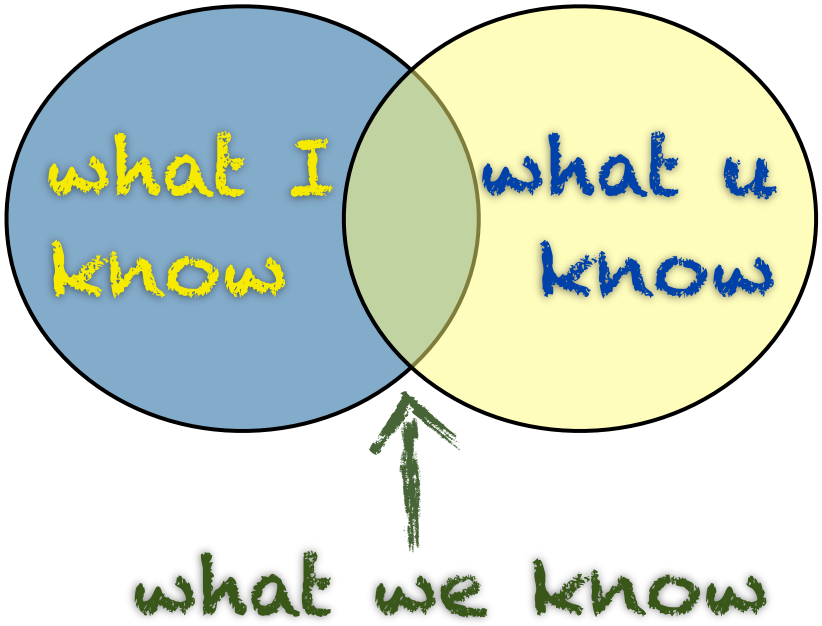
novice Δ “significantly higher”

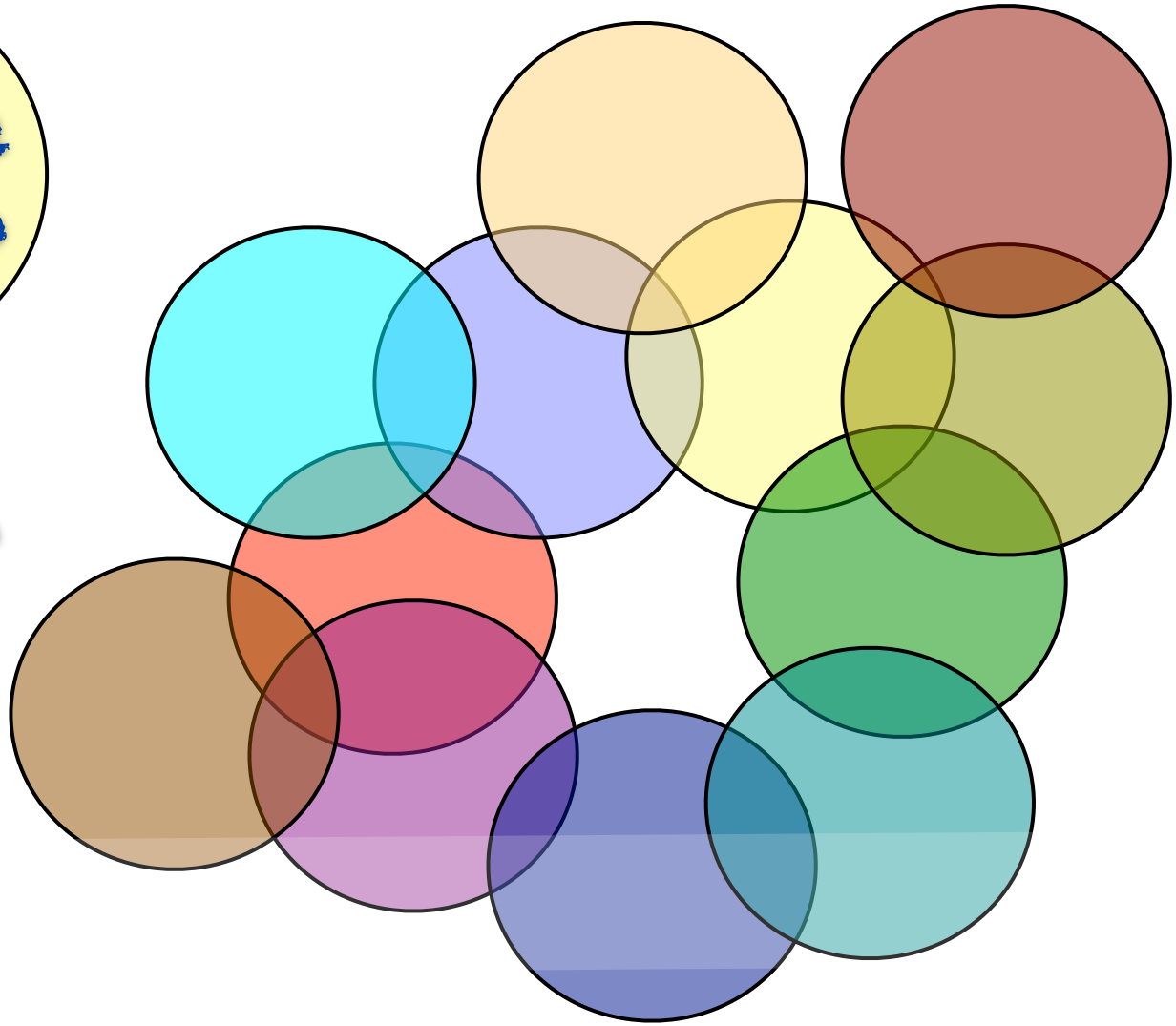
Lui, Chan, & Nosek: pairs outperform for design tasks

http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4378344

benefits

promiscuous knowledge





fungibility

domain knowledge

architectural understanding

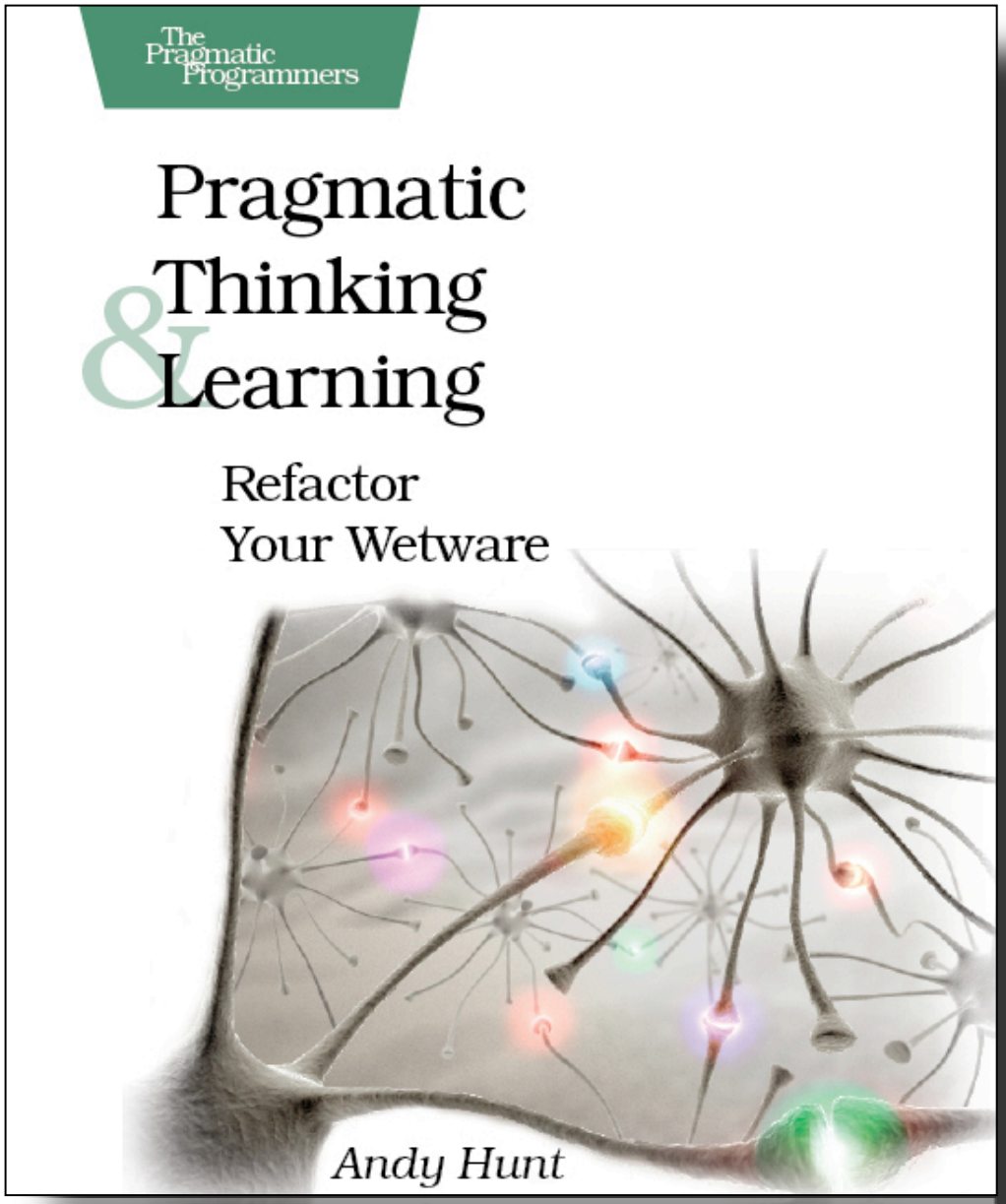
design implications

keyboard shortcuts

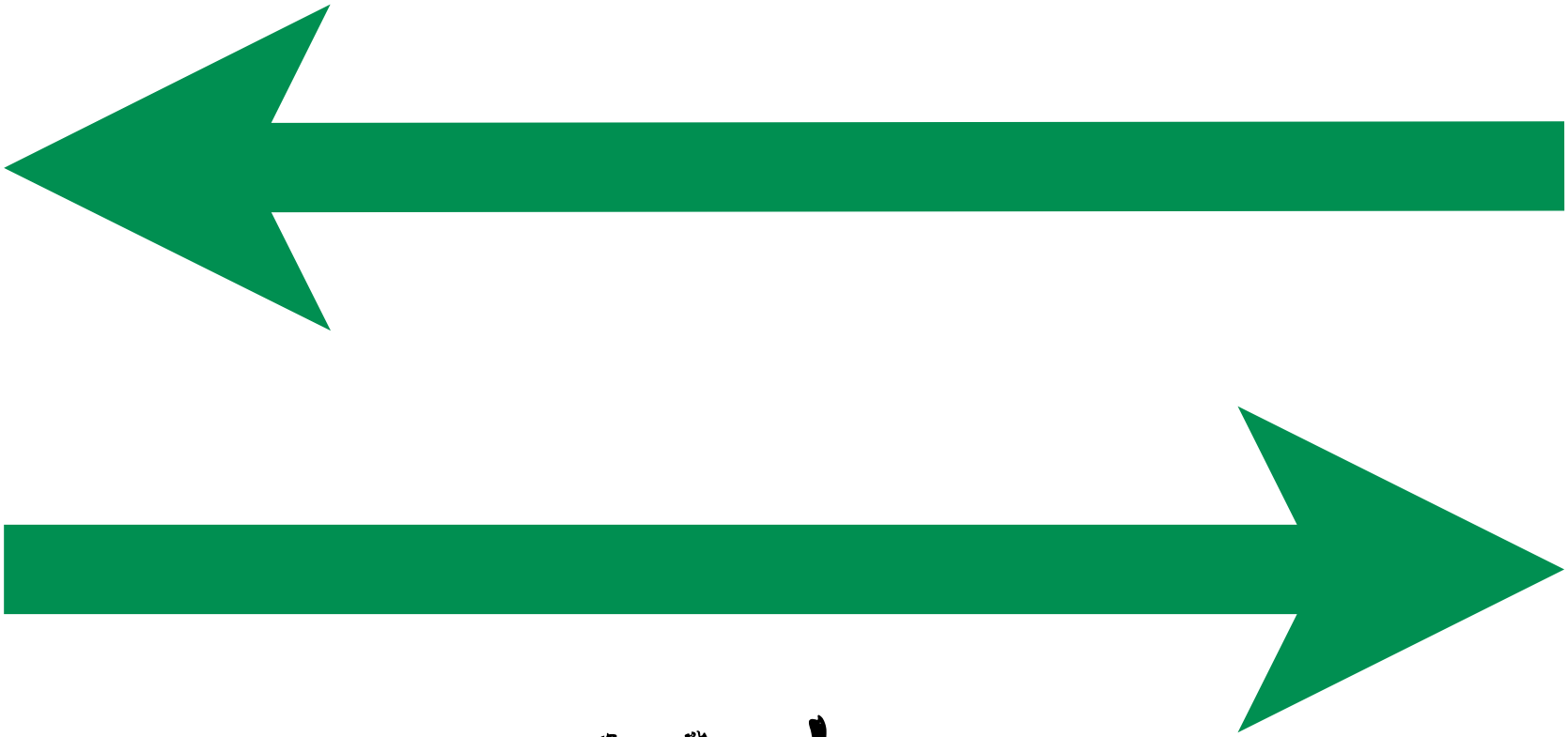
effective tools


playing w/
better players

why pair programming works


The Pragmatic Programmers
Pragmatic Thinking & Learning
Refactor Your Wetware
Andy Hunt

left brain



right brain

left brain:

spoken language and writing

counting

rational thought and logic

analysis, recognition of details

governing and lawmaking

science

awareness of time

linear thought, "step by step"

left brain

right brain

right brain

body language

ability to visualize, daydreaming

intuition

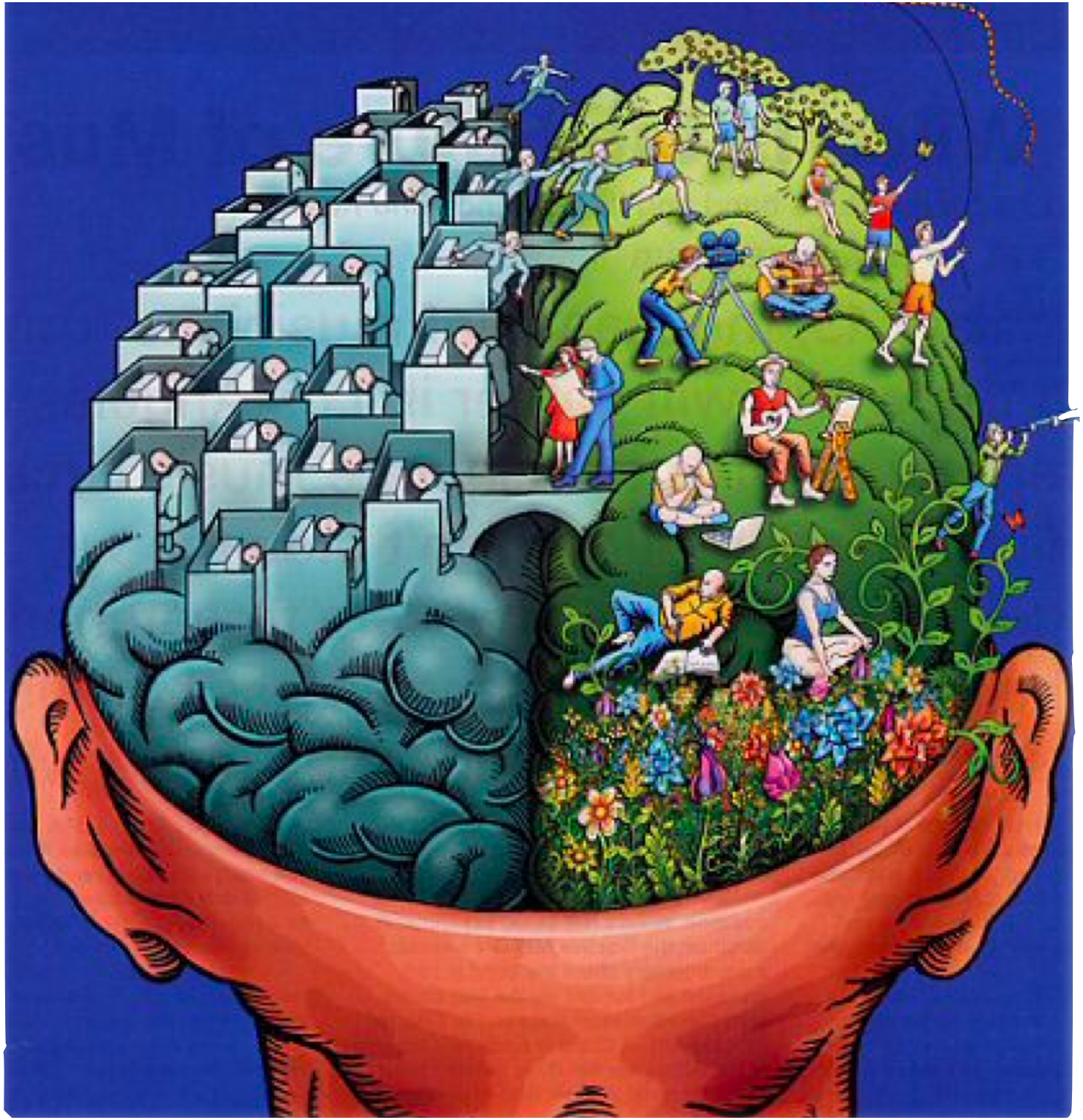
synthesis, ability to synopsize

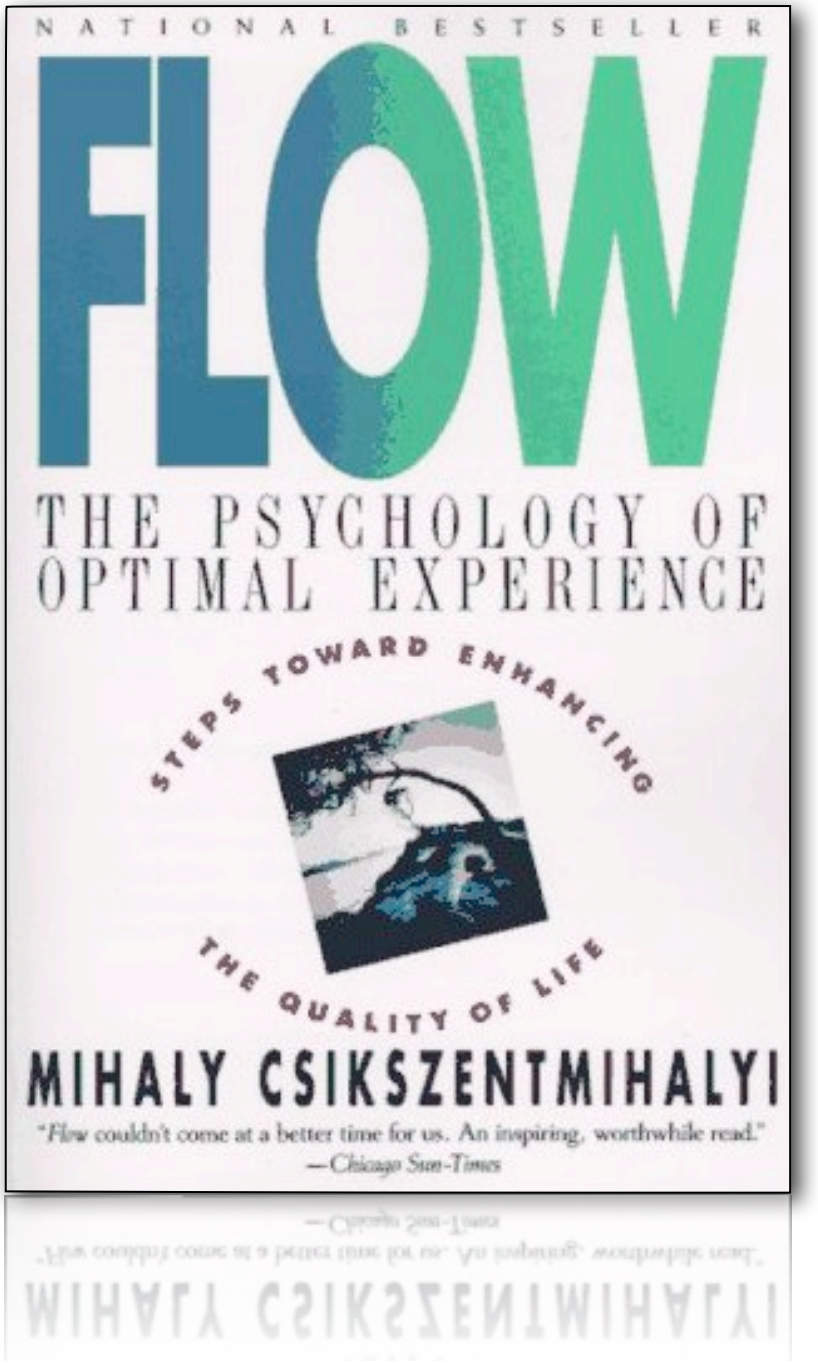
creativity, imagination

art, music, dance, color, rhythm

spacial awareness

holistic and non-linear thought




NATIONAL BESTSELLER
FLOW
THE PSYCHOLOGY OF OPTIMAL EXPERIENCE
STEPS TOWARD ENHANCING THE QUALITY OF LIFE
MIHALY CSIKSZENTMIHALYI
"Flow couldn't come at a better time for us. An inspiring, worthwhile read."
—Chicago Sun-Times

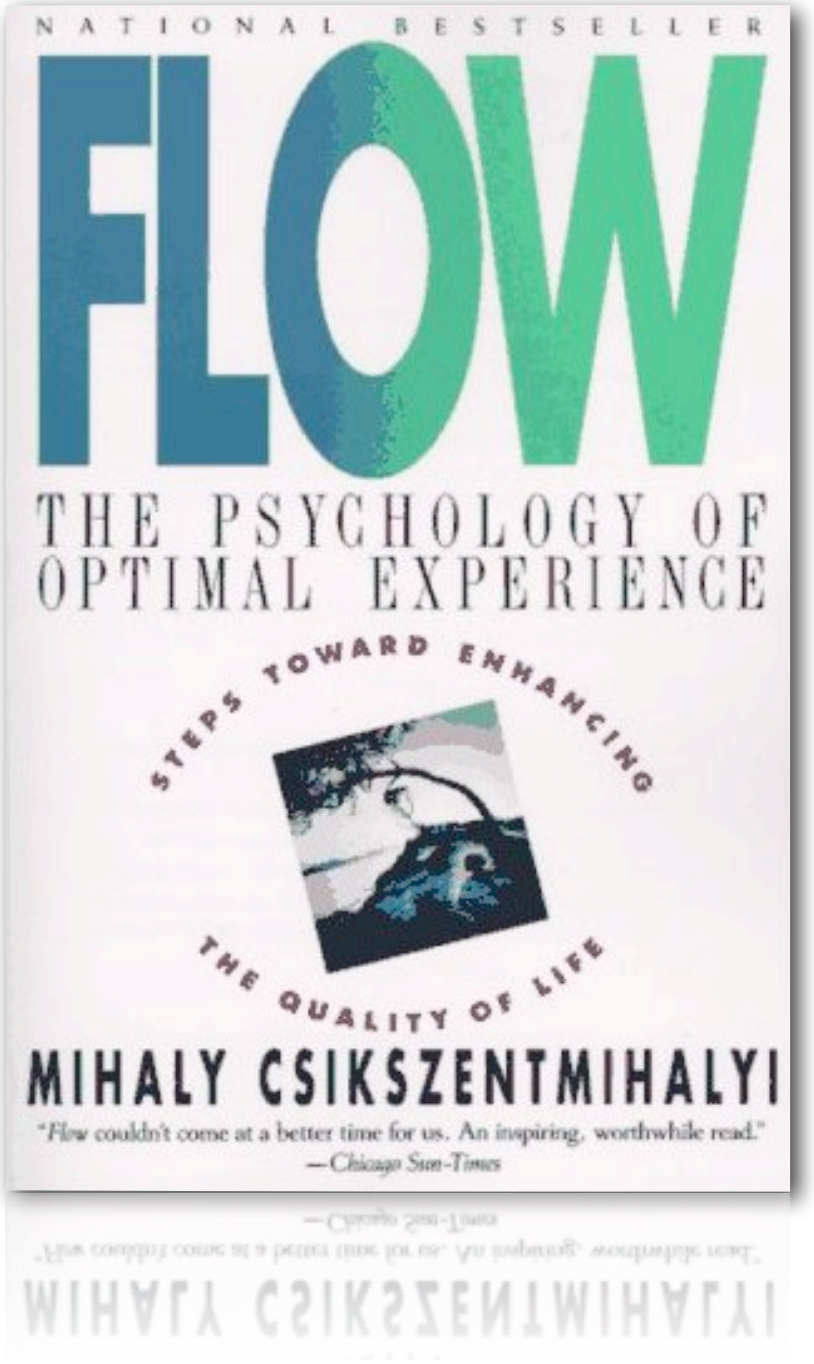
time disappears

tunnel vision

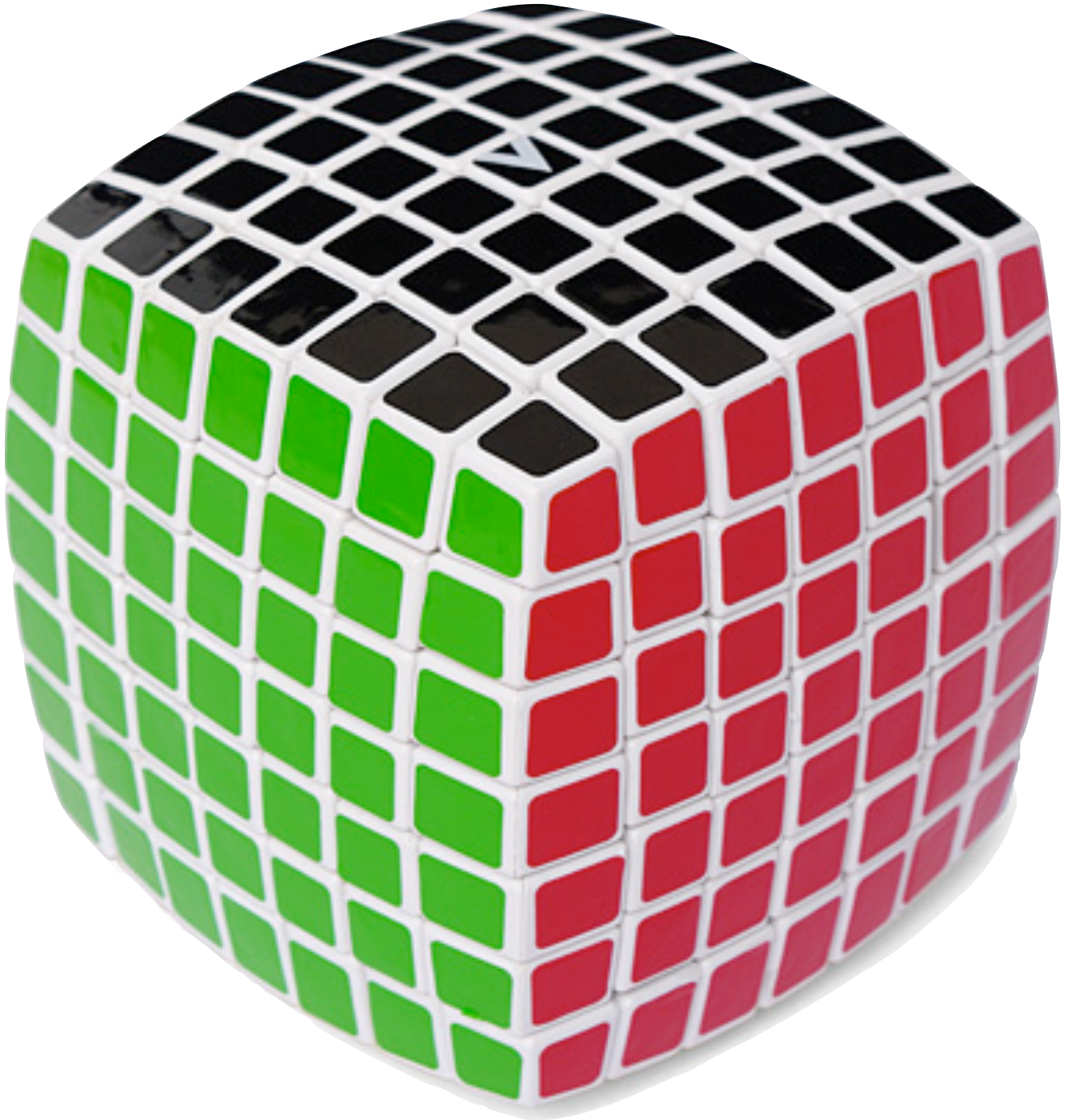
"in the zone"

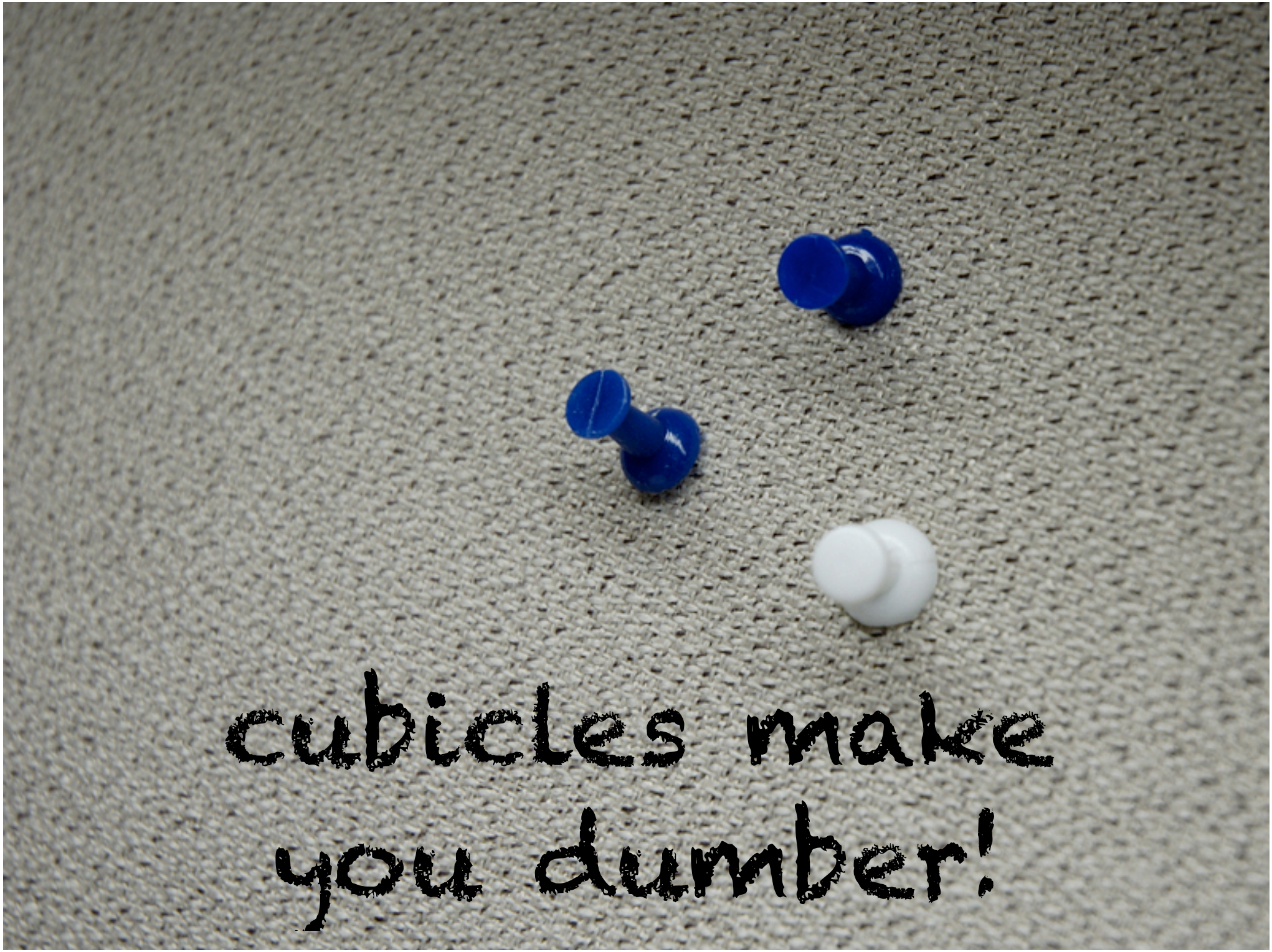
total concentration

insanely productive


NATIONAL BESTSELLER
FLOW
THE PSYCHOLOGY OF OPTIMAL EXPERIENCE
STEPS TOWARD ENHANCING THE QUALITY OF LIFE
MIHALY CSIKSZENTMIHALYI
"Flow couldn't come at a better time for us. An inspiring, worthwhile read."
—Chicago Sun-Times


Puma Productivity Pants™



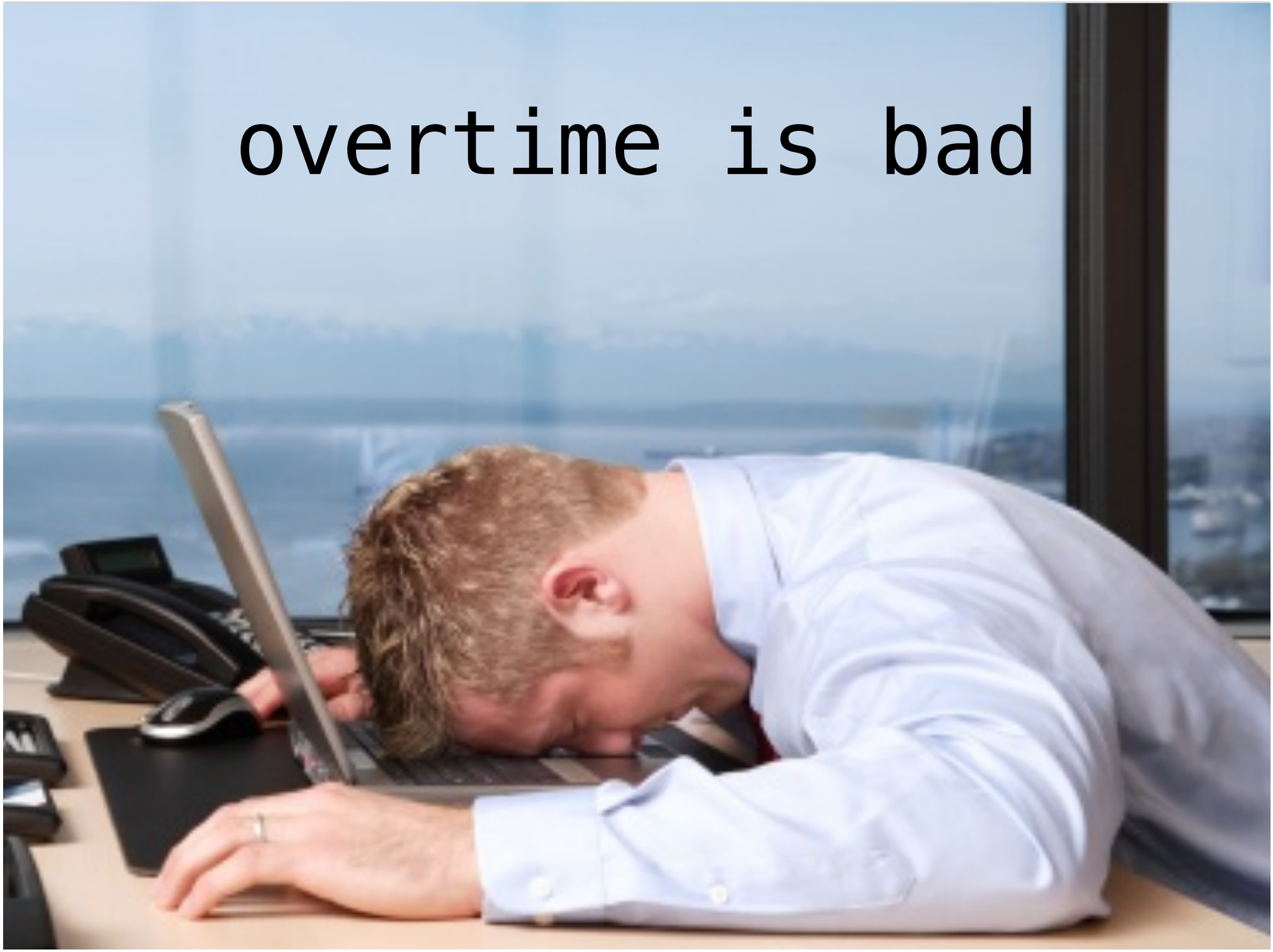

cubicles make
you dumber!


Don't Touch!

coding
!=
dull

managers

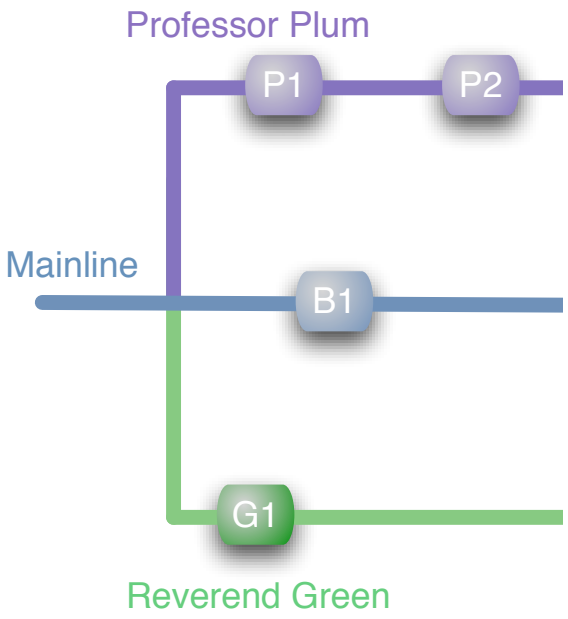
makers


overtime is bad

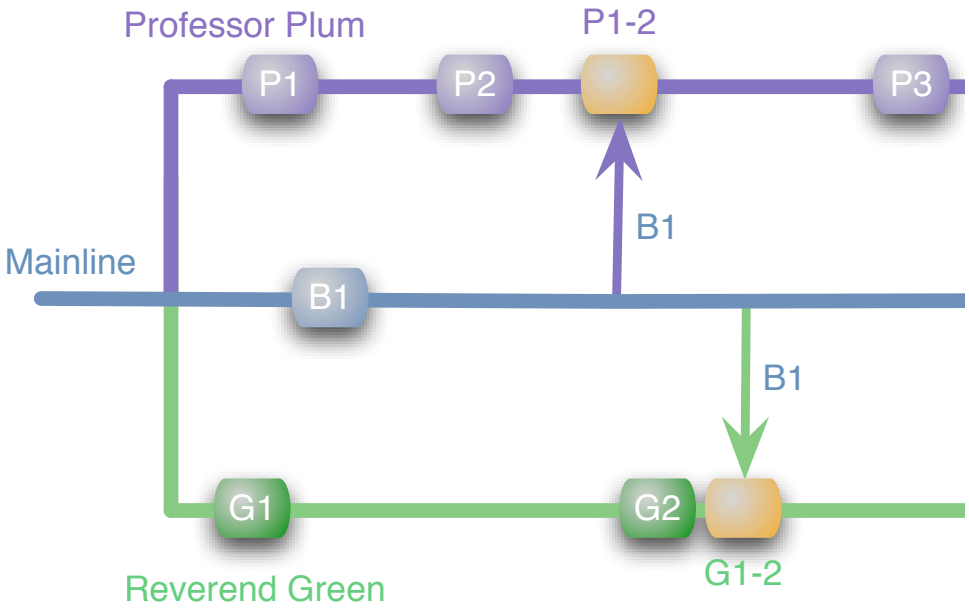
When playing with someone who is not experienced at the RPS, look out for double runs or, in other words, the same throw twice.

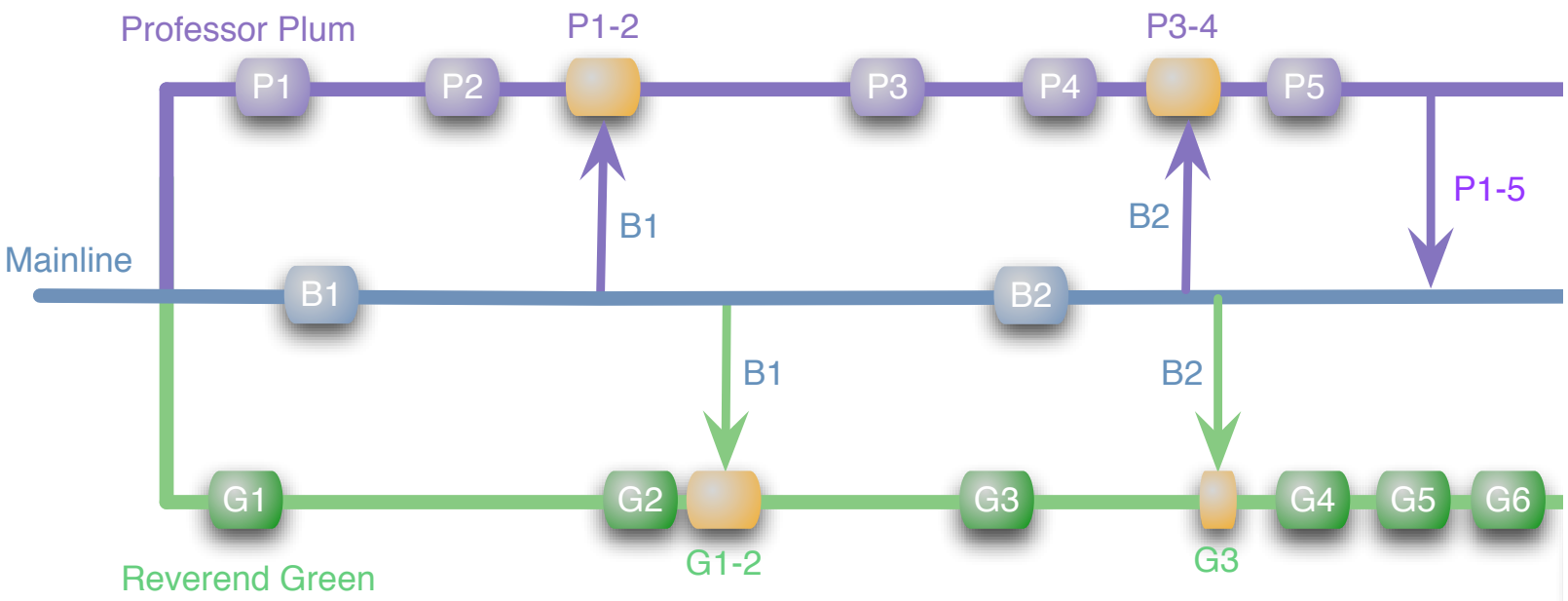
feature toggles

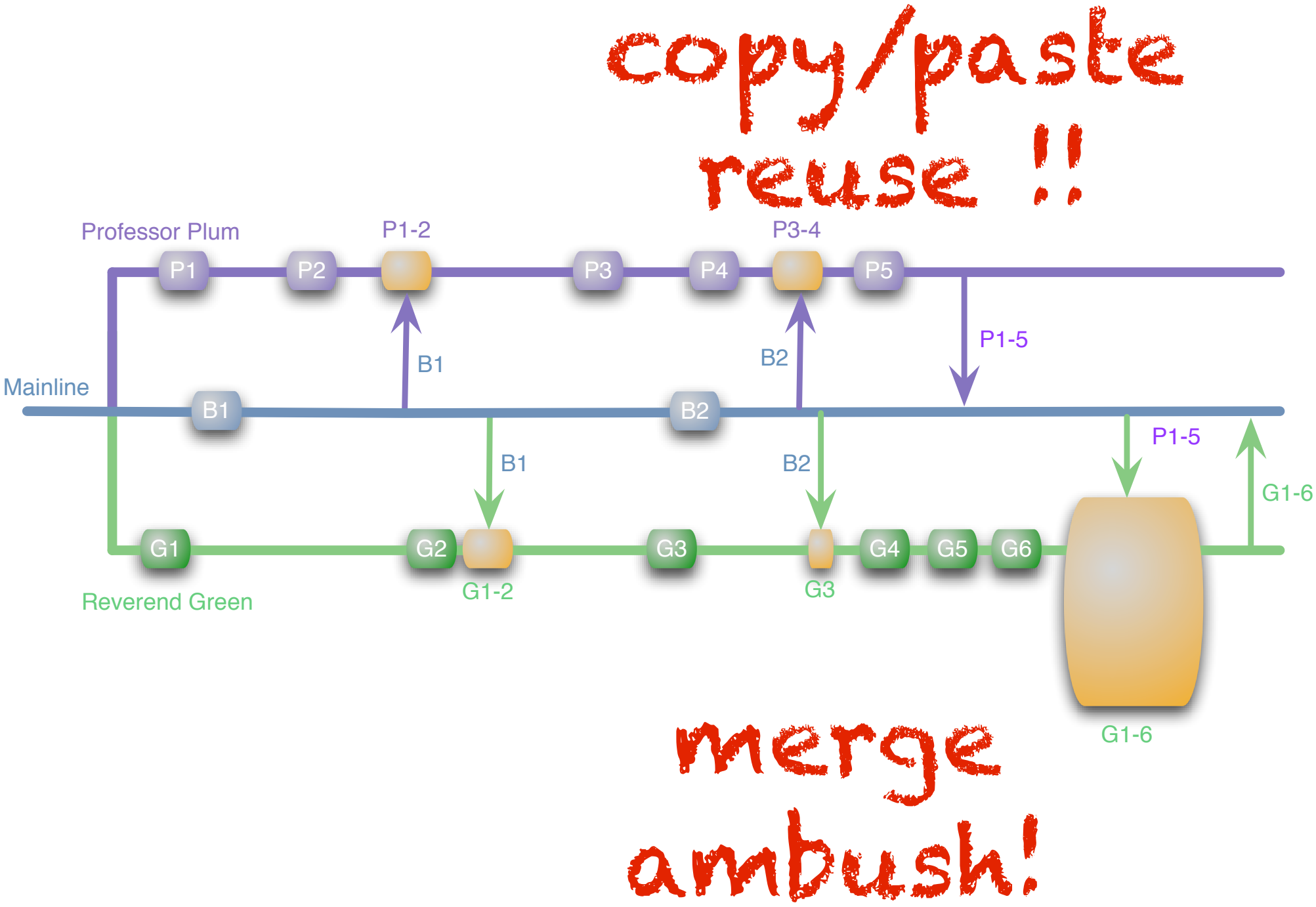
Thanks to my ThoughtWorks® colleague Cosmin Stejerean for this topic

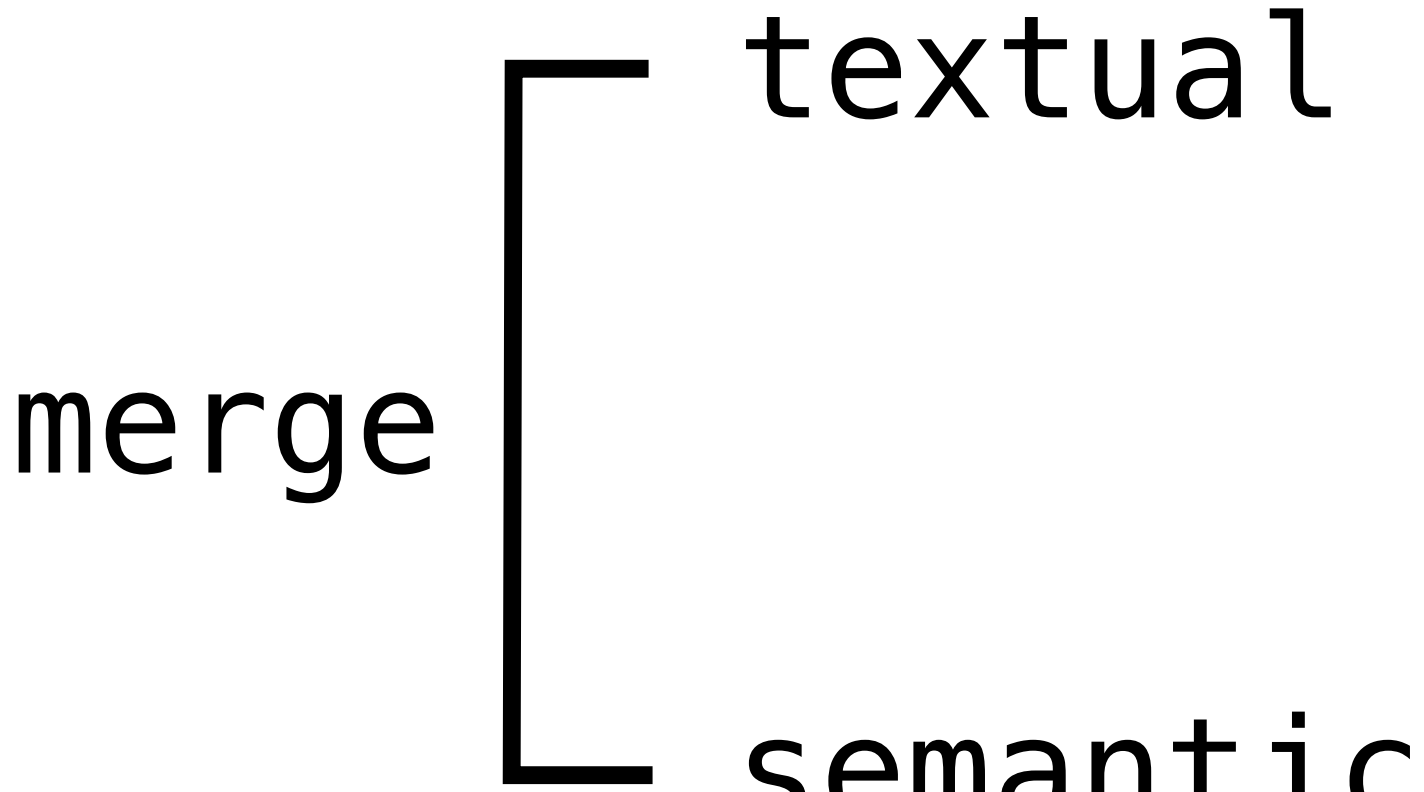

Professor Plum
P1
P2
Mainline
B1
G1
Reverend Green

feature branch




Professor Plum
P1-2
P3-4
P1
P2
P3
P4
P5
P1-5
B1
B2
Mainline
B1
B2
B1
B2
G1
G2
G3
G4
G5
G6
G1-2
G3
Reverend Green


copy/paste
reuse !!
Professor Plum
P1-2
P3-4
P1
P2
P3
P4
P5
B1
B2
P1-5
Mainline
B1
B2
B1
B2
P1-5
G1-6
G1
G2
G3
G4
G5
G6
G1-2
G3
Reverend Green
G1-6
merge
ambush!

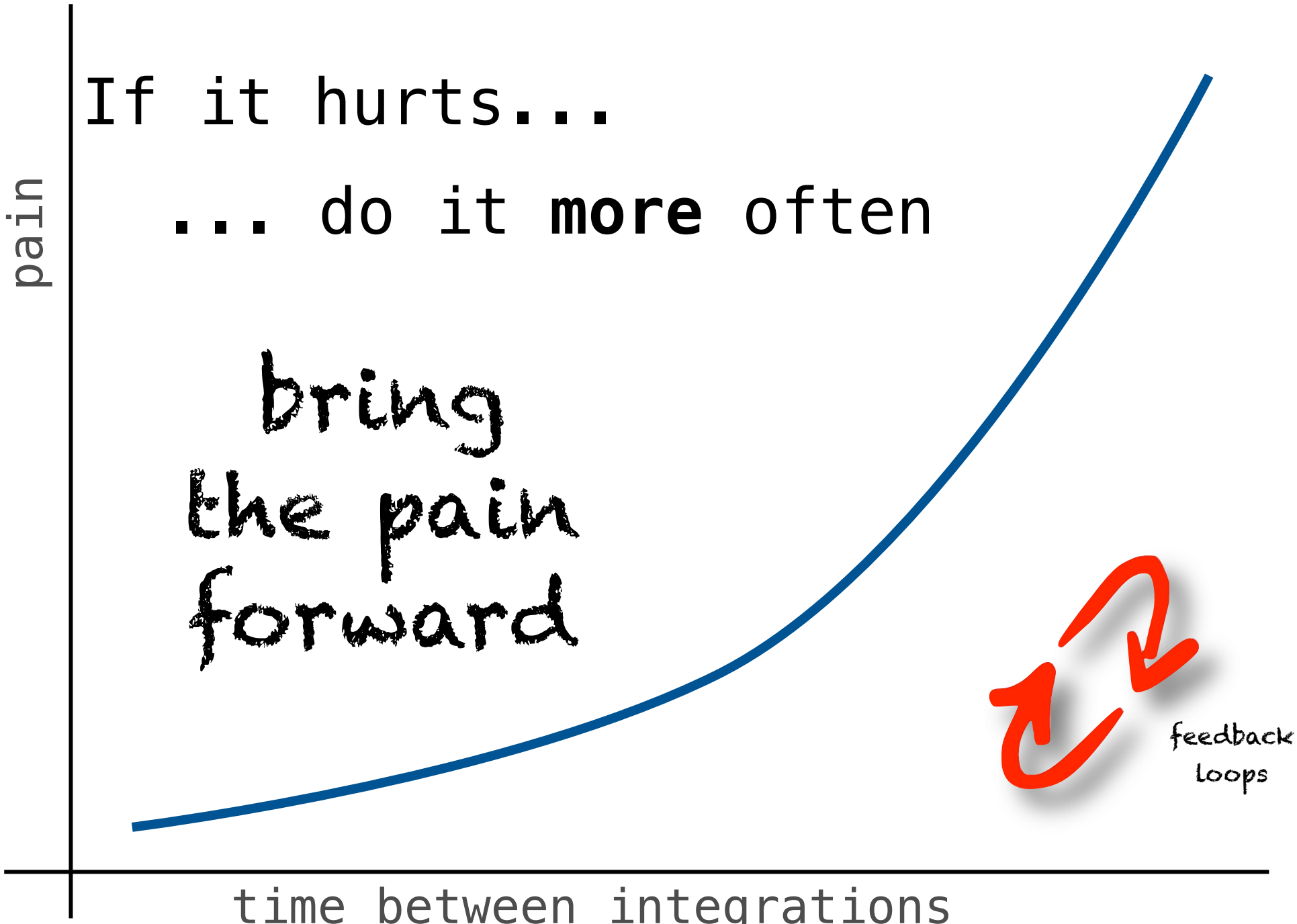


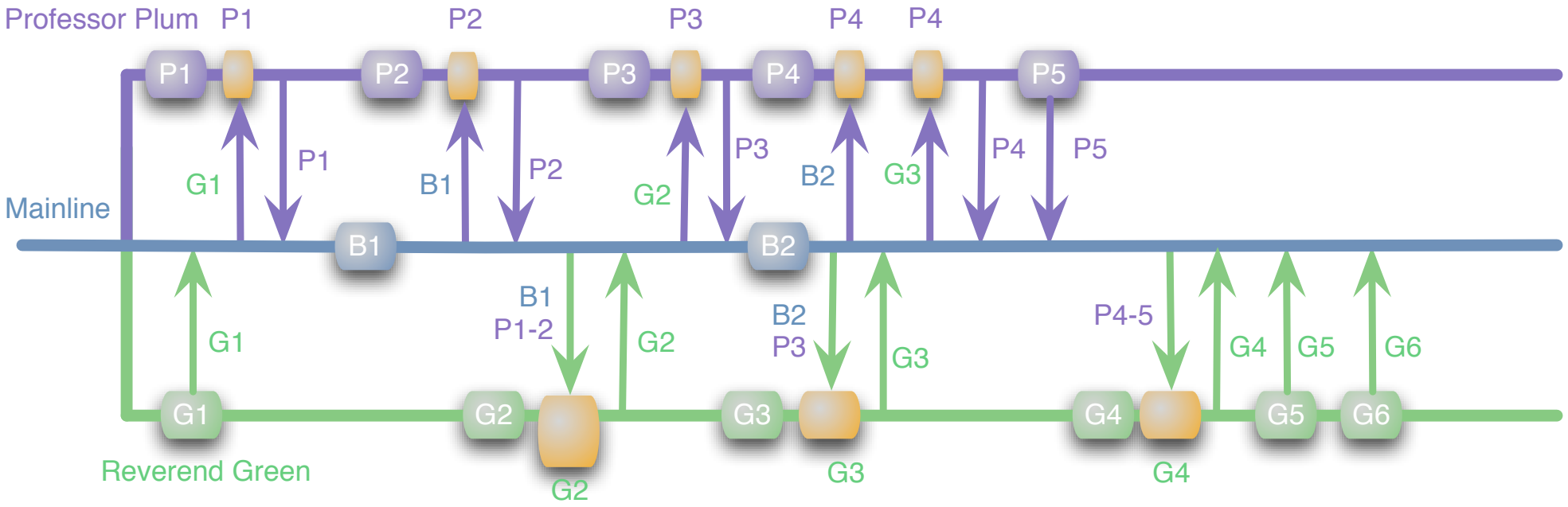
requires tests

If it hurts...

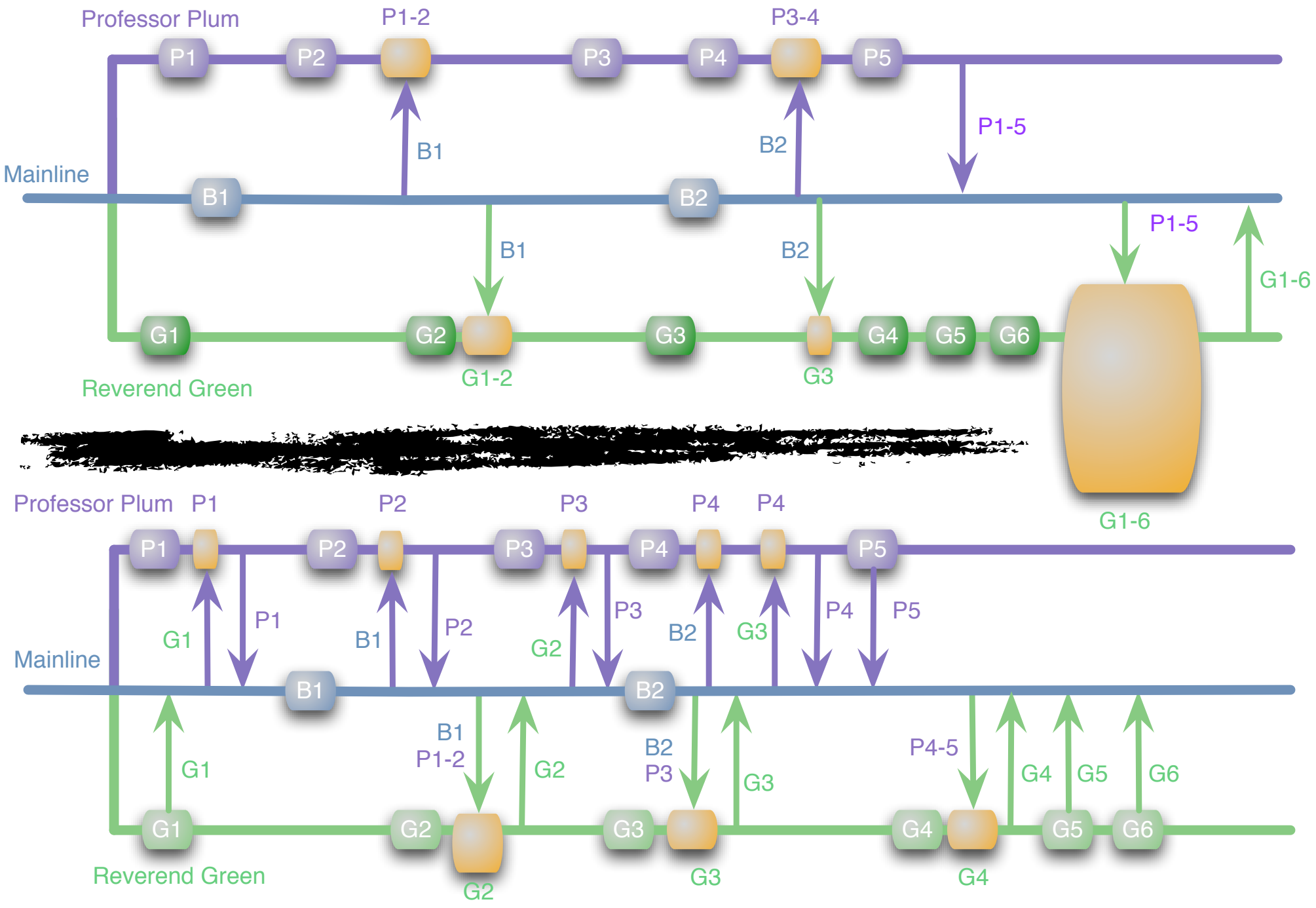
... do it **more** often

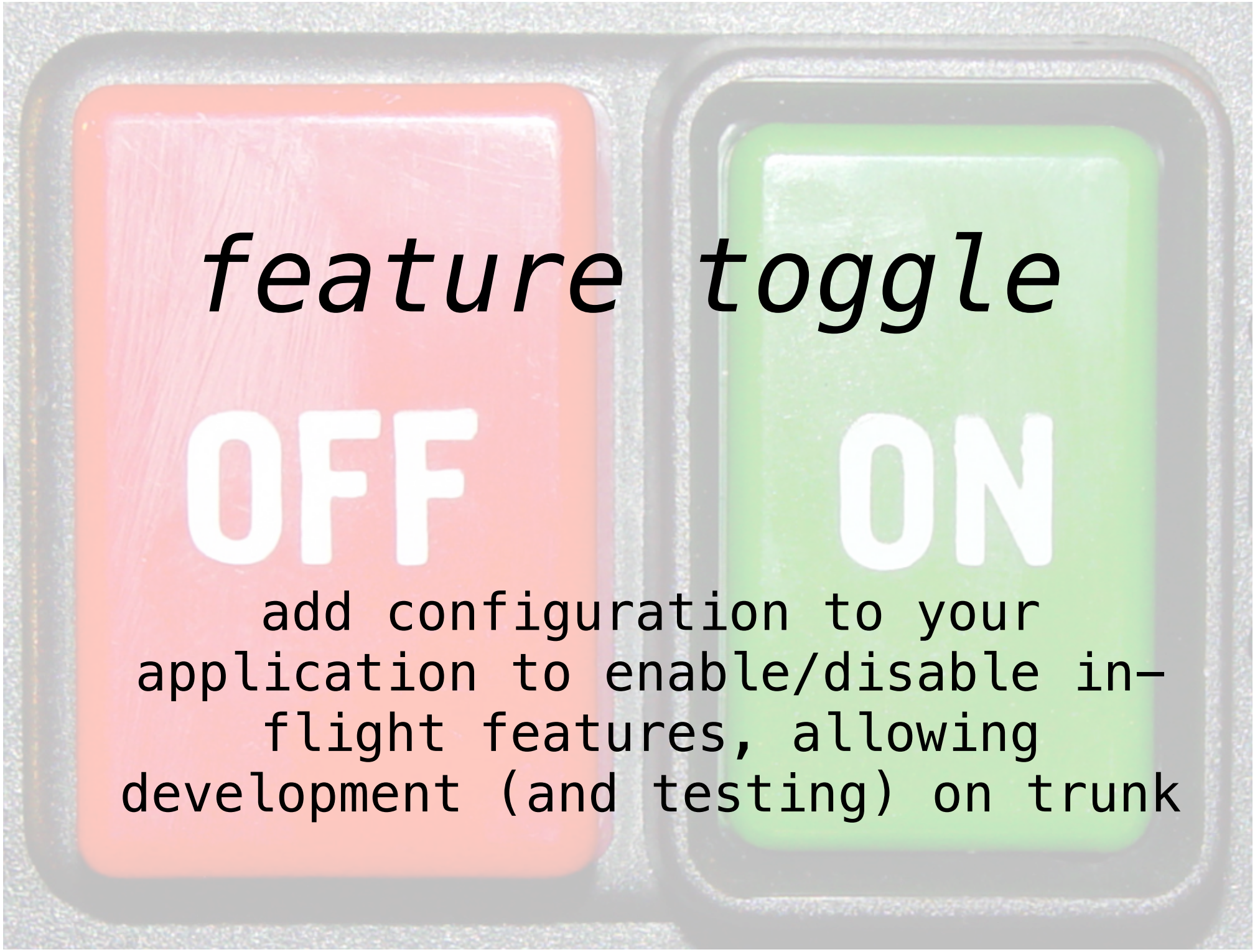
bring the pain forward





Continual Integration


Professor Plum
P1-2
P3-4
P1
P2
P3
P4
P5
B1
B2
P1-5
Mainline
B1
B2
B1
B2
P1-5
G1-6
G1
G2
G3
G4
G5
G6
G1-2
G3
Reverend Green
Professor Plum
P1
P2
P3
P4
P4
P1
P2
P3
P4
P5
G1
P1
B1
P2
G2
P3
B2
G3
P4
P5
Mainline
B1
B2
G1
B1
P1-2
G2
B2
P3
G3
P4-5
G4
G5
G6
G1
G2
G3
G4
G5
G6
Reverend Green
G2
G3
G4



feature toggle

add configuration to your application to enable/disable in-flight features, allowing development (and testing) on trunk

simple

```
<c:if test="${featureFoo}">
    <a href="/foo">Foo</a>
</c:if>
```

turn it off in the user interface

```
public void doSomething() {
    if (featureFoo) {
        «foo specific logic»
    } else {
        «regular logic»
    }
}
```

turn it off in code

inheritance

```
public interface Processor {
    void process(Bar bar);
}

public class CoreProcessor implements Processor {
    public void process(Bar bar) {
        doSomething(bar);
        handleFoo(bar);
        doSomethingElse(bar);
    }

    protected void handleFoo(Bar bar) {
    }
}

public class FooProcessor extends CoreProcessor {
    protected void handleFoo(Bar bar) {
        doSomethingFooSpecific(bar);
    }
}
```

composition

```
public interface FeatureHandler {
    void handle(Bar bar);
}

public class Processor {
    FeatureHandler handler;

    public Processor(FeatureHandler handler) {
        this.handler = handler;
    }

    public void process(Bar bar) {
        doSomething();
        handler.handle(bar);
        doSomethingElse();
    }
}

public class CoreHandler implements Handler {
    public void handle(Bar bar) {
    }
}

public class FooHandler implements Handler {
    public void handle(Bar bar) {
        doSomethingCompletelyDifferent(bar);
    }
}
```

dependency injection

annotations

```
@Retention(RetentionPolicy.RUNTIME)
public @interface Foo {
    boolean value() default true;
}
```

```
@Foo(false) public class CoreProcessor implements Processor {
«»
}
@Foo public class FooProcessor extends CoreProcessor {
«»
}
```

```
public class FeatureIncludeFilter implements TypeFilter {

    private final TypeFilter fooFilter = new AnnotationTypeFilter(Foo.class, true);

    public boolean match(MetadataReader metadataReader,
                         MetadataReaderFactory metadataReaderFactory)
        throws IOException {

        if (fooFilter.match(metadataReader, metadataReaderFactory)) {
            boolean value = getAnnotationValue(metadataReader, Foo.class);

            if (FeatureToggles.isFooEnabled()) {
                return value;
            } else {
                return !value;
            }
        }
        return false;
    }

    private boolean getAnnotationValue(MetadataReader metadataReader,
                                       Class annotationClass) {
        return (Boolean) metadataReader.
            getAnnotationMetadata().
            getAnnotationAttributes(annotationClass.getName()).
            get("value");
    }
}
```

```
<context:component-scan base-package="com.example.features">
    <context:include-filter type="custom"
        expression="com.example.features.FeatureIncludeFilter" />
</context:component-scan>
```

```
public interface Processor {

}

@Foo(false)
public class CoreProcessor implements Processor {

}

@Foo
public class FooProcessor extends CoreProcessor {

}
```

separating static assets

leave static assets as static files

create feature-specific versions of the static content

include conditionally into dynamic templates

shopping_cart.css

shopping_cart_foo.css

build vs runtime

build-time toggles:

- never leak details
- builds only what's released

run-time toggles:

- long-lived feature toggles
- more flexible testing

cleaning up

remove feature toggles once feature becomes official

exception: multiple versions

don’t featurize your application to death

continuous delivery

When playing against someone who asks you to remind them about the rules, take the opportunity to subtly "suggest a throw" as you explain to them by physically showing them the throw you want them to play.

continuous integration

integrate early & often

deployment

deploy as the final stage of CI

delivery

software is always deployable

principles

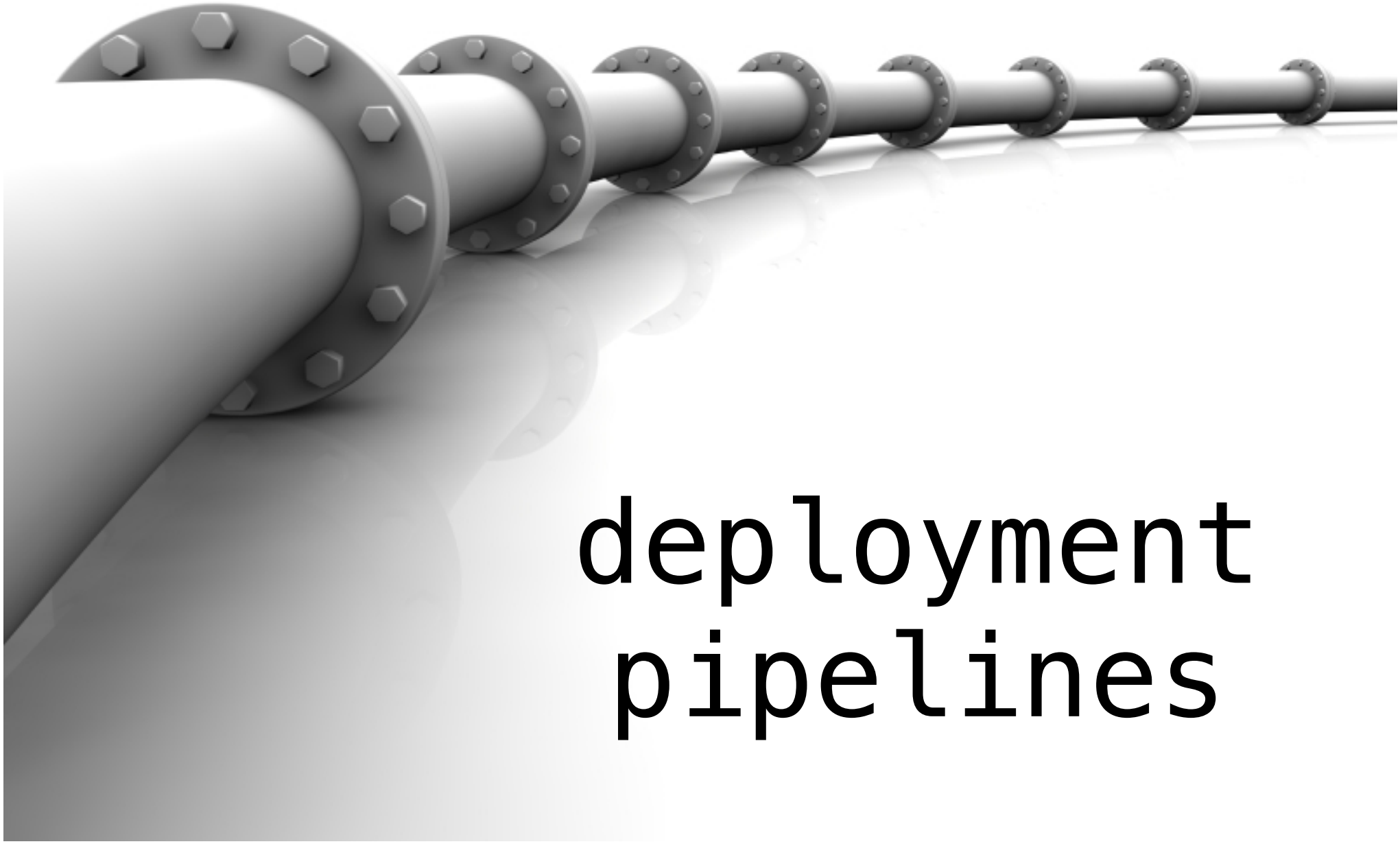
create a repeatable, reliable process for releases

automate almost everything

keep everything in version control

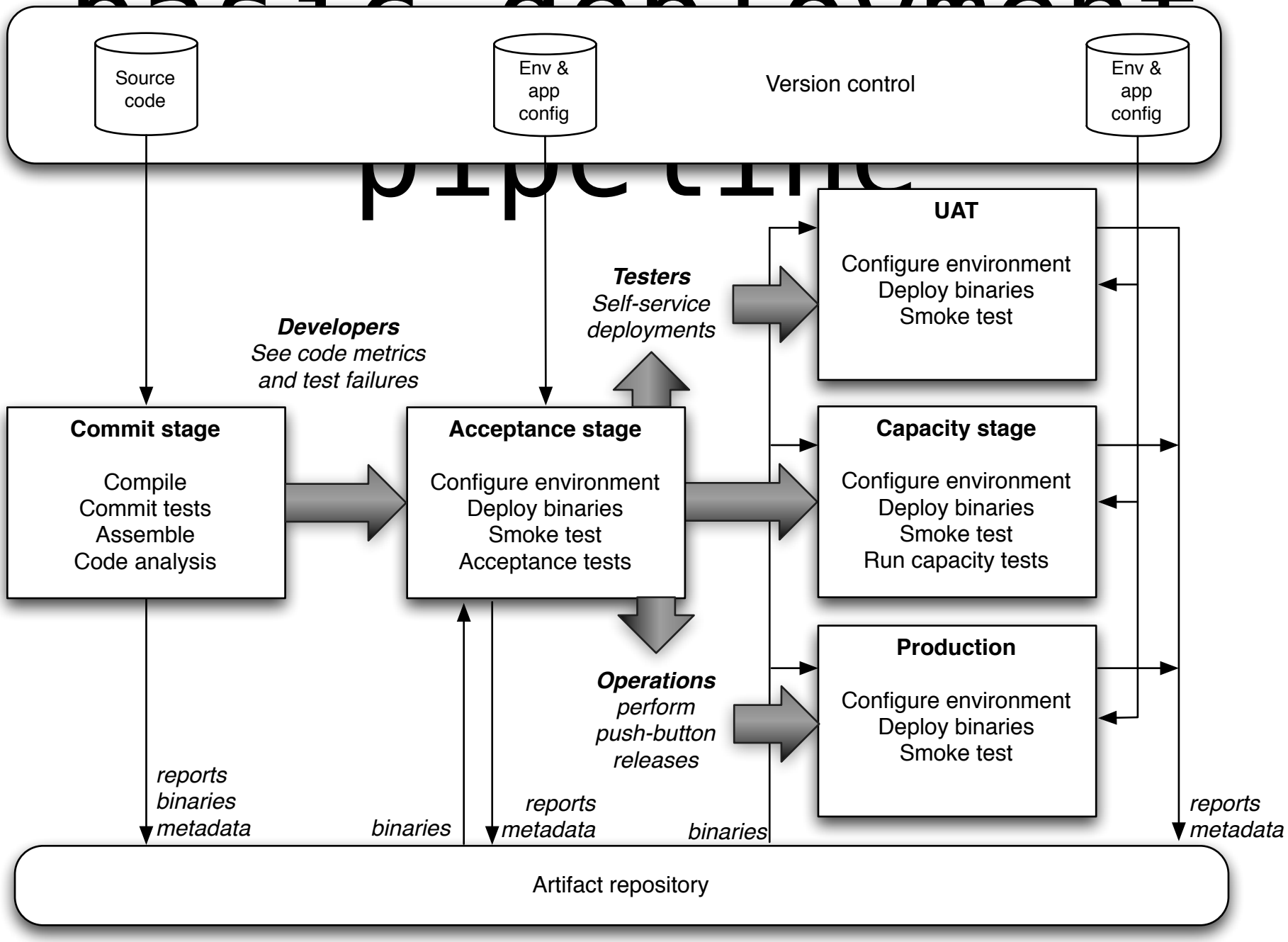
if it hurts, do it more frequently

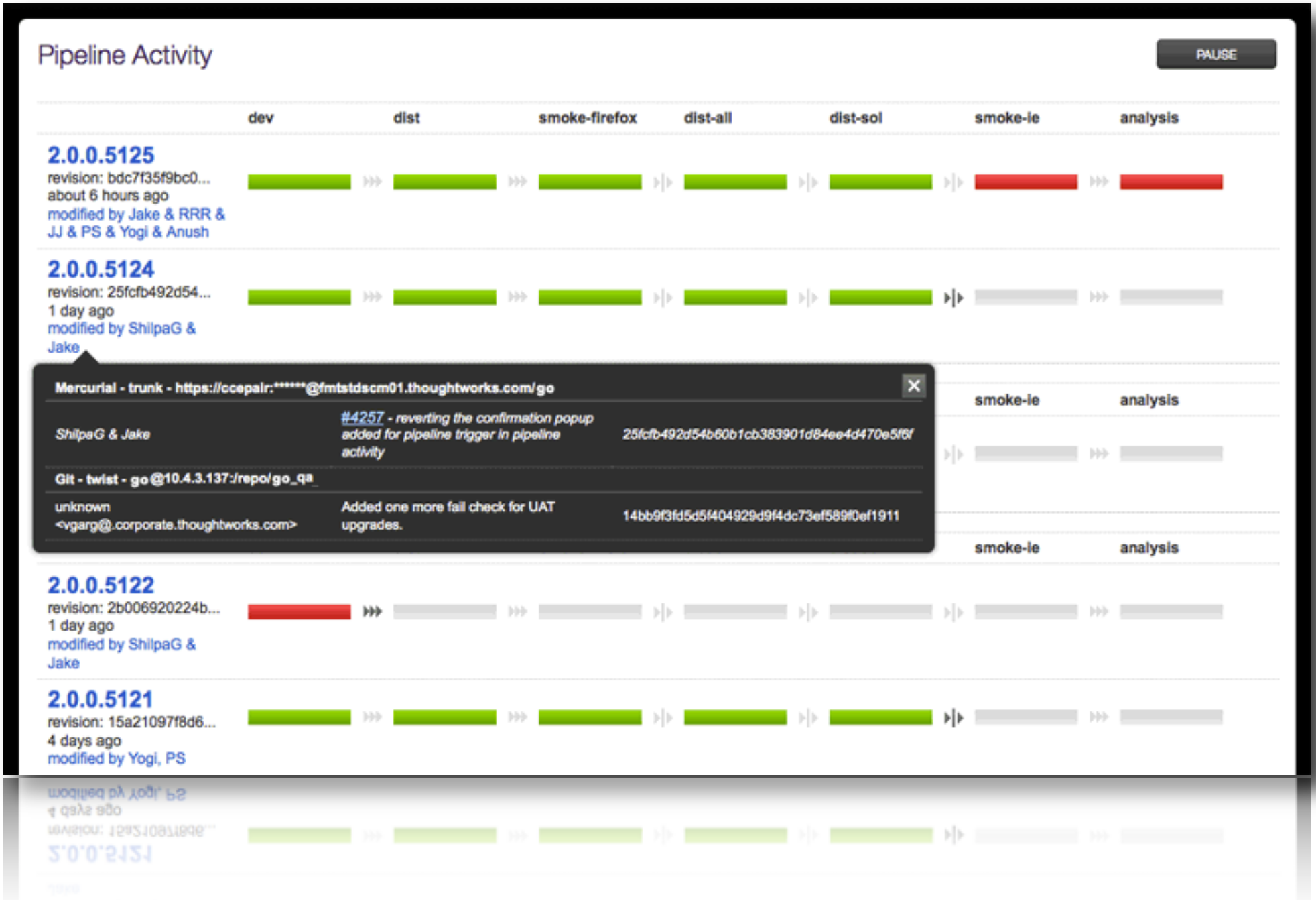
“done” means “released”

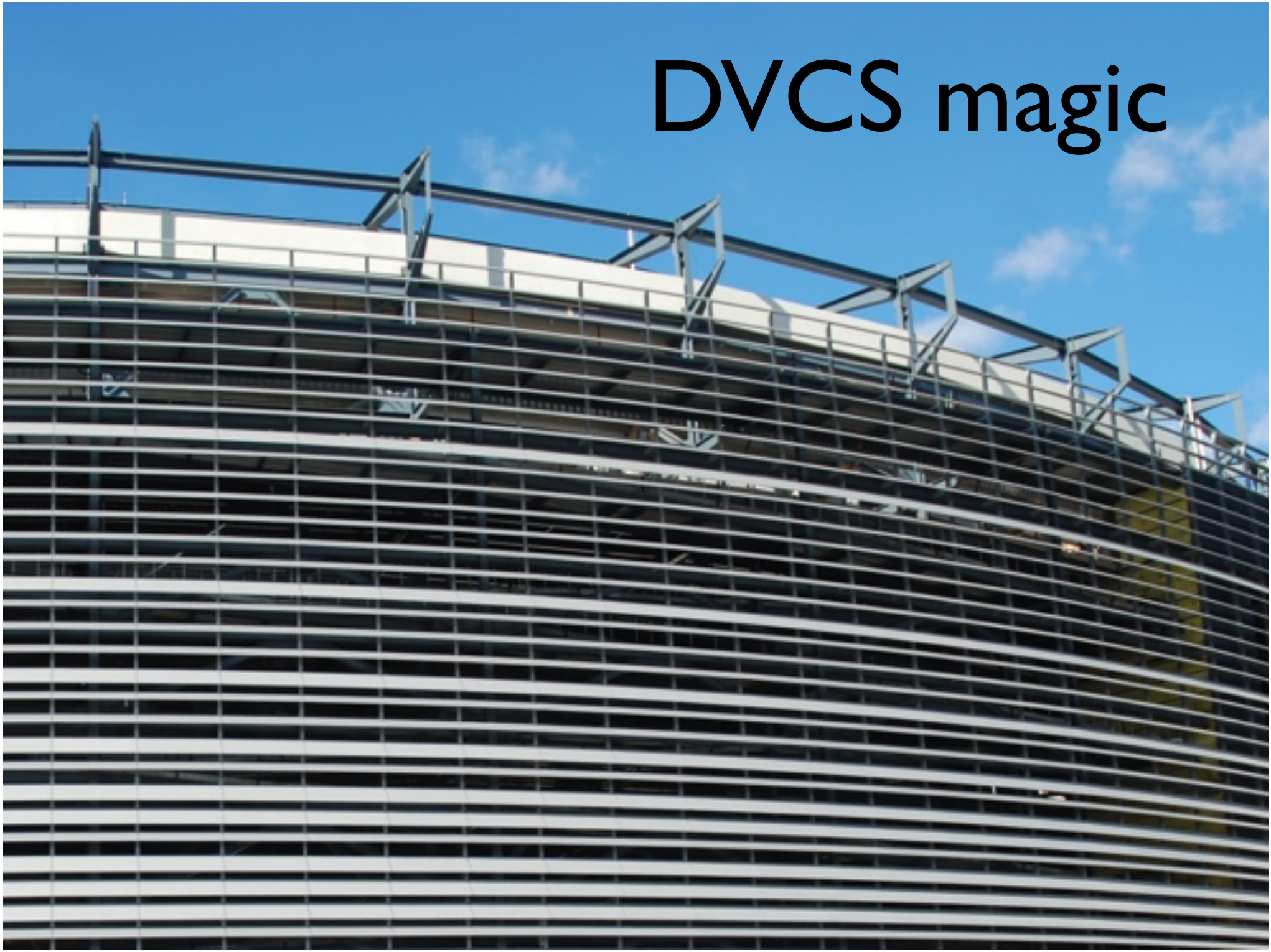


deployment pipelines

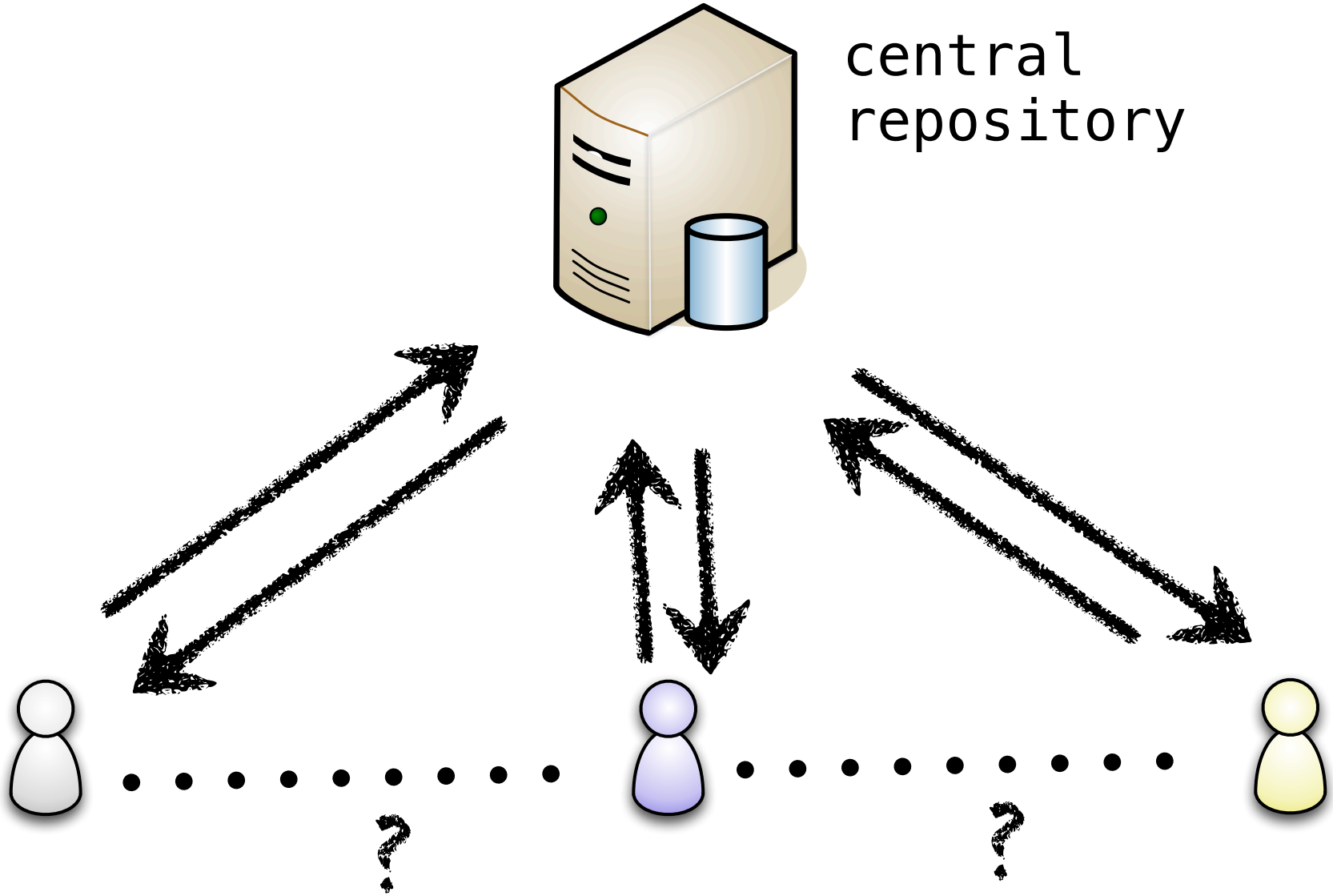
basic deployment pipeline



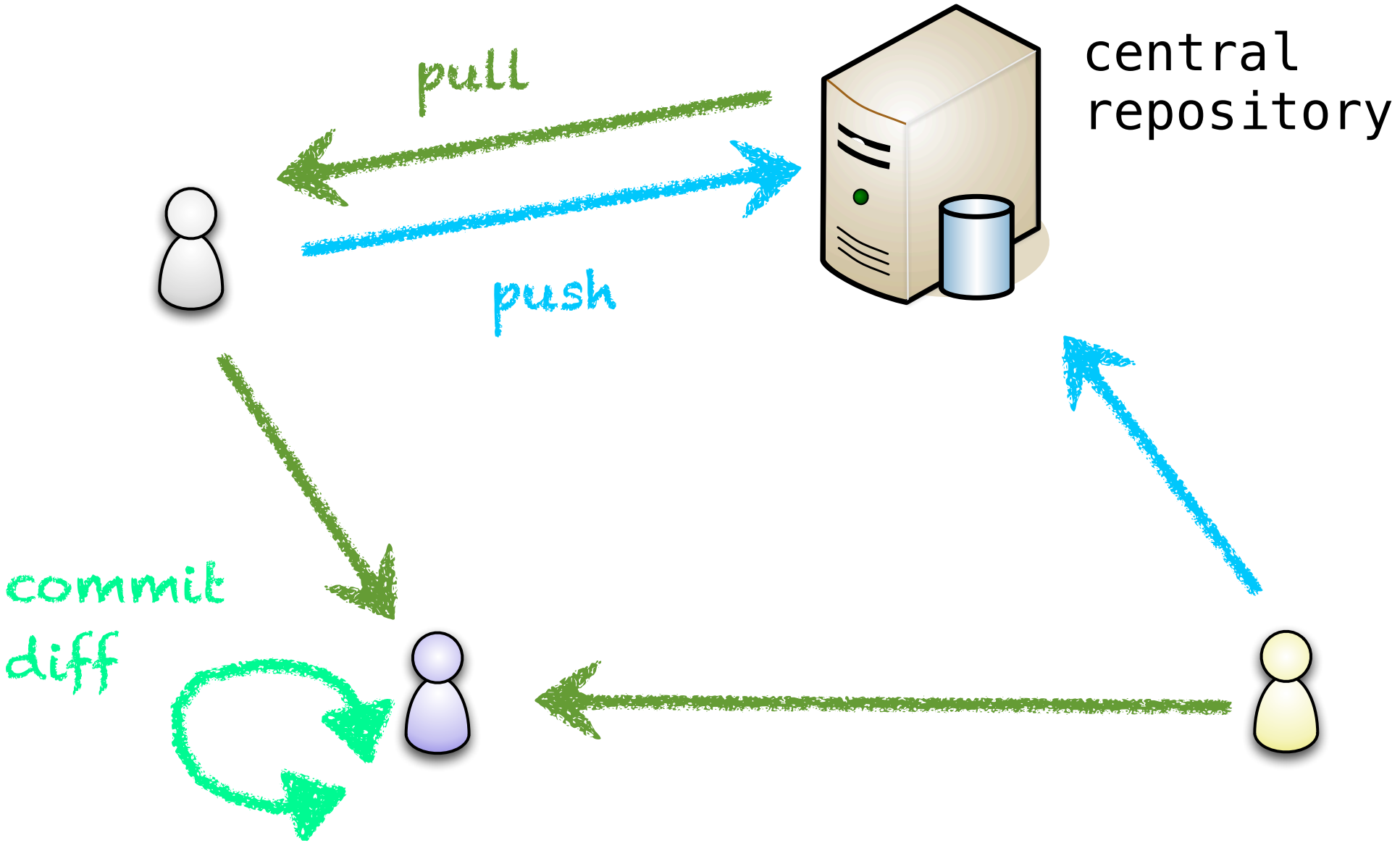

Pipeline Activity
PAUSE
dev
dist
smoke-firefox
dist-all
dist-sol
smoke-ie
analysis
2.0.0.5125
revision: bdc7f35f9bc0...
about 6 hours ago
modified by Jake & RRR & JJ & PS & Yogi & Anush
2.0.0.5124
revision: 25fcfb492d54...
1 day ago
modified by ShilpaG & Jake
Mercurial - trunk - https://ccepalr:******@fmtstdscm01.thoughtworks.com/go
ShilpaG & Jake
#4257 - reverting the confirmation popup added for pipeline trigger in pipeline activity
25fcfb492d54b60b1cb383901d84ee4d470e5f6f
Git - twist - go@10.4.3.137:/repo/go_qa
unknown <vgarg@.corporate.thoughtworks.com>
Added one more fail check for UAT upgrades.
14bb9f3fd5d5f404929d9f4dc73ef589f0ef1911
2.0.0.5122
revision: 2b006920224b...
1 day ago
modified by ShilpaG & Jake
2.0.0.5121
revision: 15a21097f8d6...
4 days ago
modified by Yogi, PS


DVCS magic

centralized VCS



DE-centralized VCS




git magic #1

svn workflow

finish feature

svn up

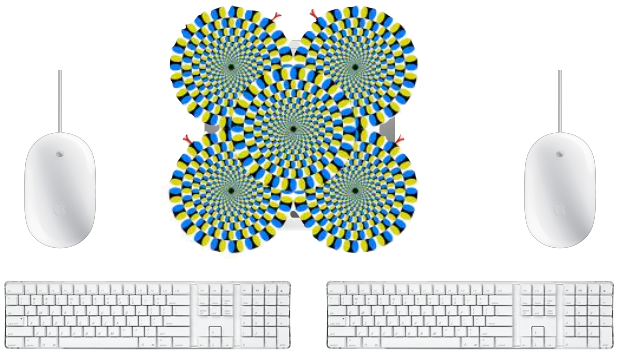
run tests locally

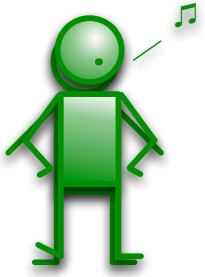
kick off checkin bash script

wait...

svn server

continuous integration server





1. pull from svn

2. run local tests

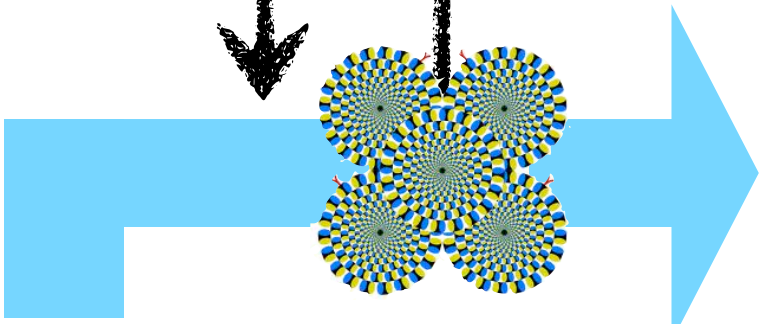
3. check in

10 min / pair / check-in

git server

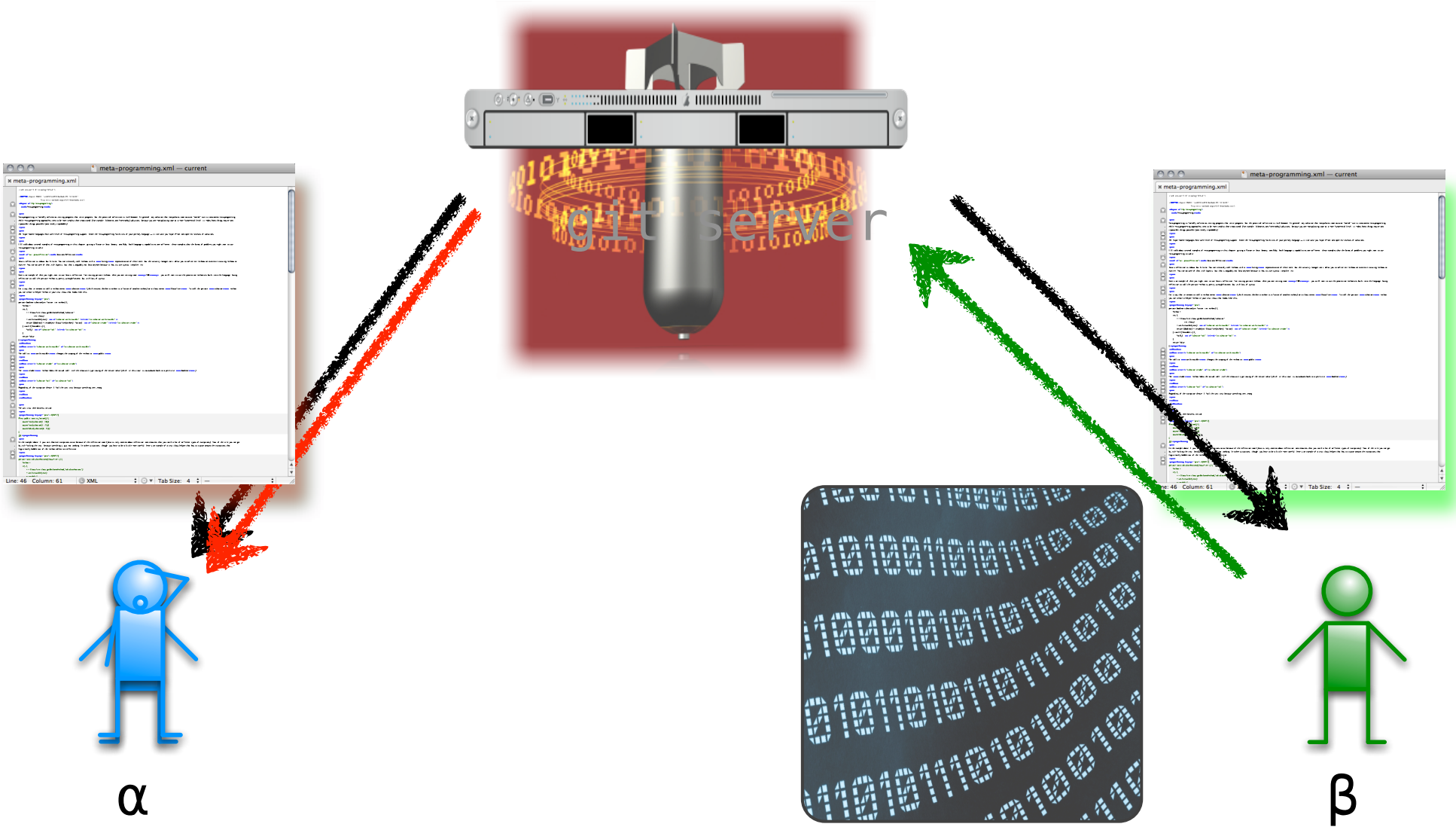
feedback
loops

continuous
integration
server



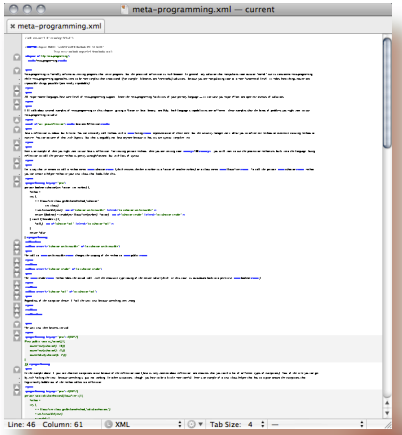
1. spawn local branch
2. pull from server
3. run tests
4. check in OR stash
5. kill branch

git magic #2

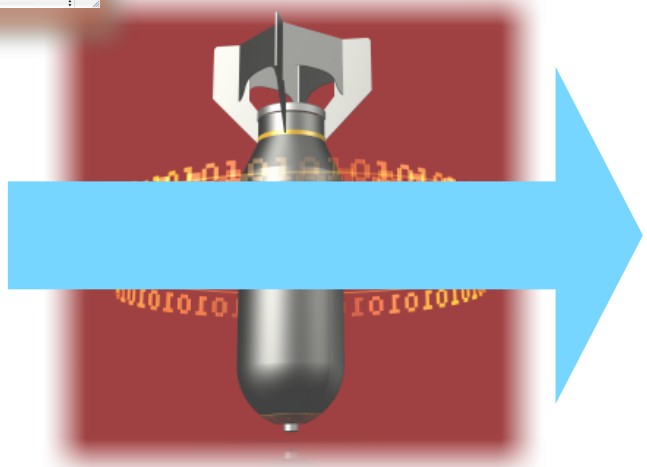


git magic #2

git server



1. undo disastrous checkout
2. save changes to local stash



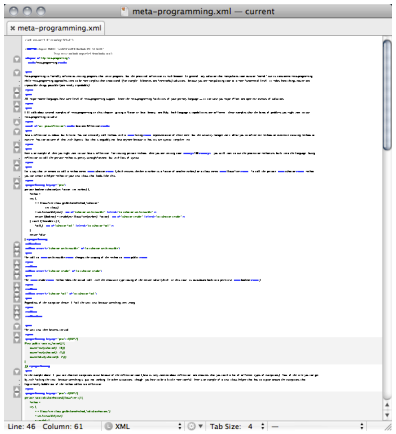
3. create local branch
4. push stash to local branch

α

β

git magic #2

git server



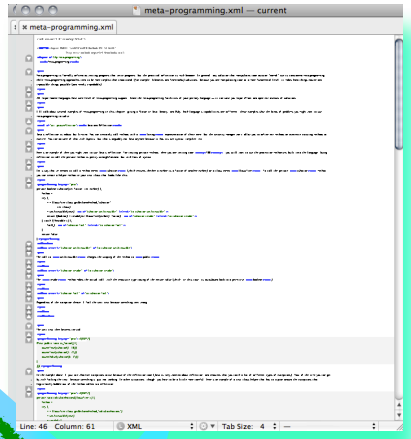
5. push local branch to remote branch

α

β

git magic #2

git server



6. stash recent changes
7. checkout remote branch

α

8. fix it!
9. check into main
10. unstash & get back to work

β

automation

time & space

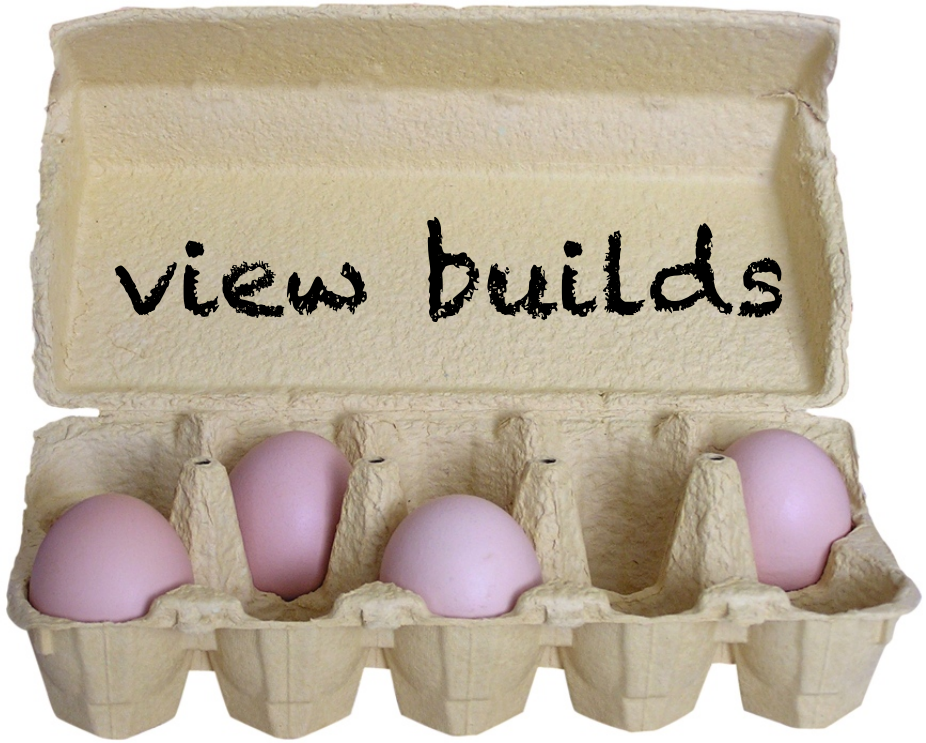
transfer a merge conflict to the person better qualified to fix it.

non-

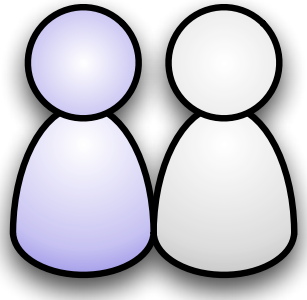
intuitivity

demonstration


why all the
rochambeau?



01_trunk_commit | 02_trunk_acceptance | 03_trunk_apache
04_trunk_externals | 05_trunk_metrics | 07_trunk_qa_tests
11_release_commit | 12_release_acceptance | 13_release_apache
14_release_externals | 17_release_qa_tests | 97_deploy_ba
98_deploy_staging | 99_spider_production | ove-search-infrastructure
in-service | ove-core-trunk | ove-core-release | ove-datasets
ove-externals | ove-externals-trunk | ove-query-counts
webservices-core | z-deploy-ba-trunk | z-deploy-endeca-ba-trunk
z-deploy-iqa-release | z-deploy-sqa-trunk | *ove-view-trunk*
ove-view-release-branch



worst ...job ...ever

http://www.worldrps.com/

?'s

please fill out the session evaluations

NEAL FORD software architect / meme wrangler

ThoughtWorks®

nford@thoughtworks.com
3003 Summit Boulevard, Atlanta, GA 30319
www.nealford.com
www.thoughtworks.com
blog: memeagora.blogspot.com
twitter: neal4d

resources

XProgramming.com - Ron Jeffries site
http://xprogramming.com/

Dr. Laurie Williams
http://collaboration.csc.ncsu.edu/laurie/publications.html

git branching model
http://nvie.com/git-model

Extreme Programming: A Gentle Introduction
http://www.extremeprogramming.org/

http://martinfowler.com/articles/continuousIntegration.html

- Maintain a Single Source Repository.
- Automate the Build
- Make Your Build Self-Testing
- Everyone Commits To the Mainline Every Day
- Every Commit Should Build the Mainline on an Integration Machine
- Keep the Build Fast
- Test in a Clone of the Production Environment
- Make it Easy for Anyone to Get the Latest Executable
- Everyone can see what's happening
- Automate Deployment