let's **move**
the **java world**

Extreme Programming practices for your team

Paweł Lipiński

geecon 2013

whoami

- ~15 years as a developer, ~11 years in Java
- programming, consulting, training, auditing, architecturing, coaching, team leading
- Formal & agile methods
- currently: developer, coach, ceo @ PRAGMATISTS

geecon 2013

Nokia Test

- iterations timeboxed and < 4 weeks
- newly created features tested and working at the end of each iteration
- iteration must start before its specification is complete

That's not agile yet, we're only talking about being iterative here.

Nokia Test - SCRUM

- you know who the PO is
- there is a Product Backlog prioritised by business value
- the Product Backlog has estimates created by the Team
- the Team maintains burndown charts and knows their velocity
- there are no PMs disrupting the work of the Team

Once upon a time...

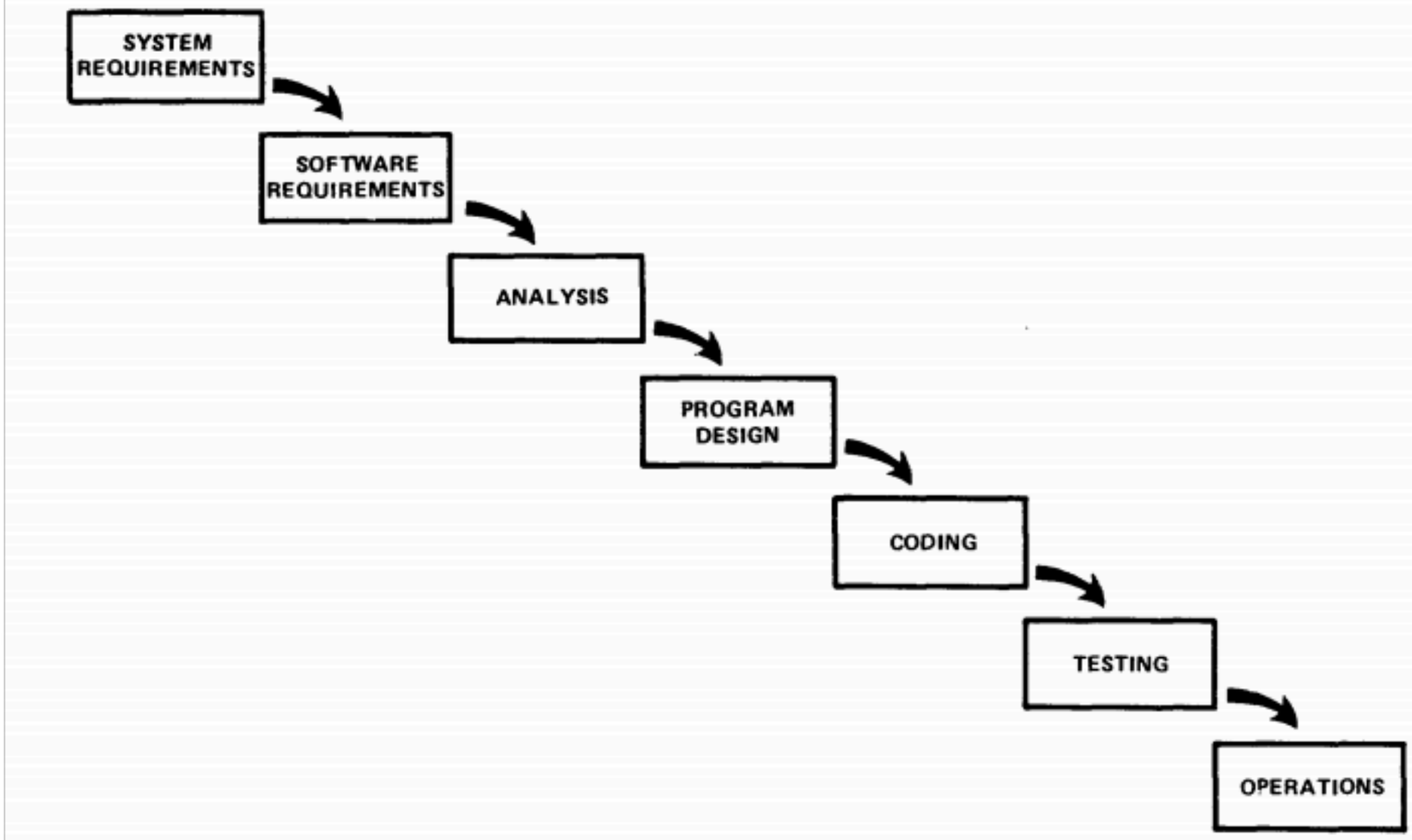


Figure 2. Implementation steps to develop a large computer program for delivery to a customer.

1970. Royce, Winston (1970), "Managing the Development of Large Software Systems",
Proceedings of IEEE WESCON **26** (August): 1–9.

He was the first who described the Waterfall model for software development, although Royce did not use the term "waterfall" in that article,
nor advocated the waterfall model as a working methodology.
Royce in fact said that it was **"risky and invites failure"** and went on to describe Incremental development. DOD-STD-2167 coined the term Waterfall to refer to the diagram on page 2 of Dr. Royce's paper.

http://en.wikipedia.org/wiki/Winston_W._Royce

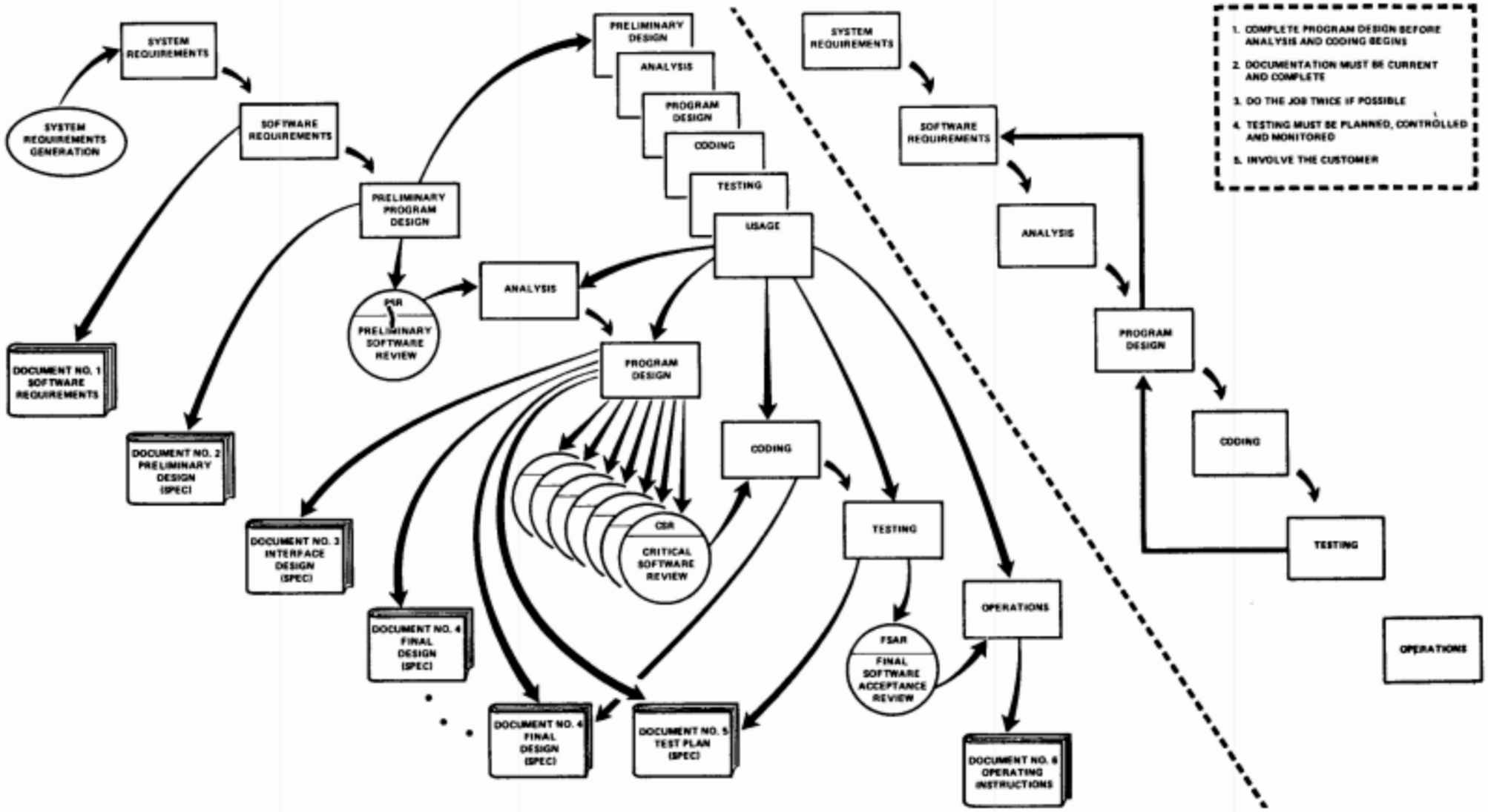


Figure 10. Summary

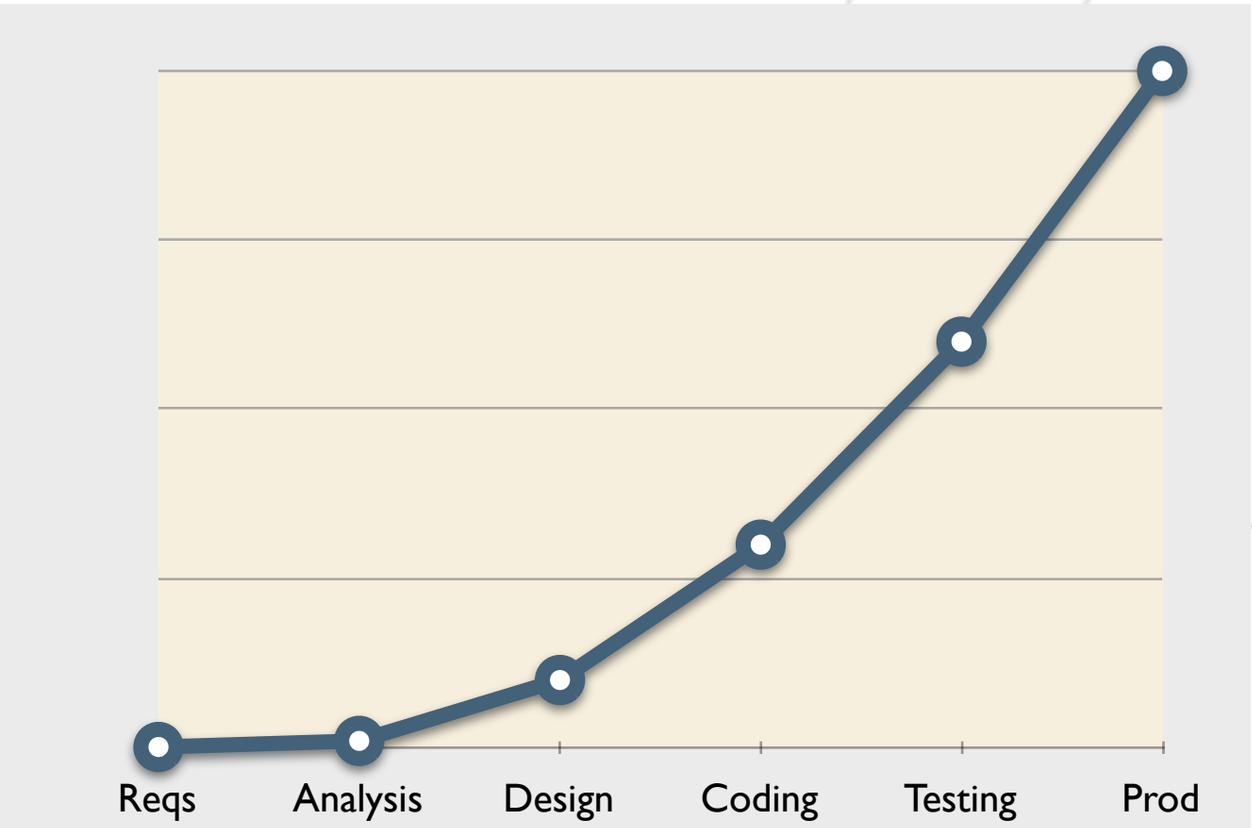
1970. Royce, Winston (1970), "Managing the Development of Large Software Systems",
Proceedings of IEEE WESCON **26** (August): 1–9.

Cost of change grows exponentially with time

Barry Boehm

Ideas are cheaper to change than code.

Bugs found early are cheaper to fix.



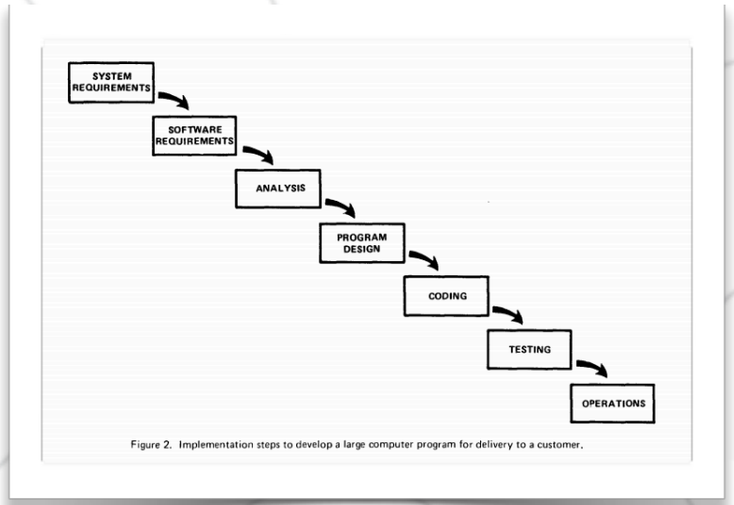
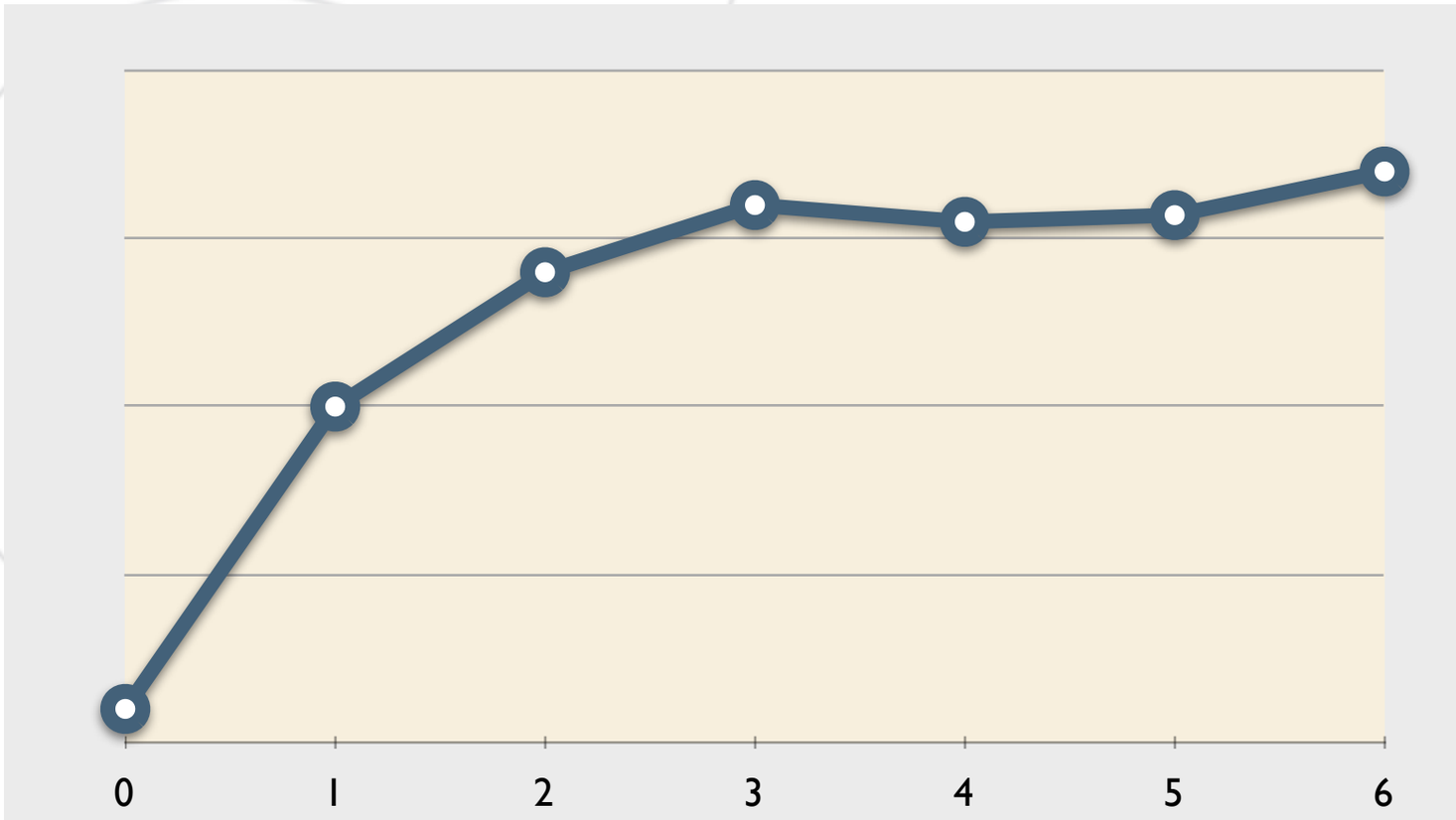


Figure 2. Implementation steps to develop a large computer program for delivery to a customer.

Does cost of change really grow exponentially with time?

Postponing decisions and reducing feedback cycles flattens the curve.



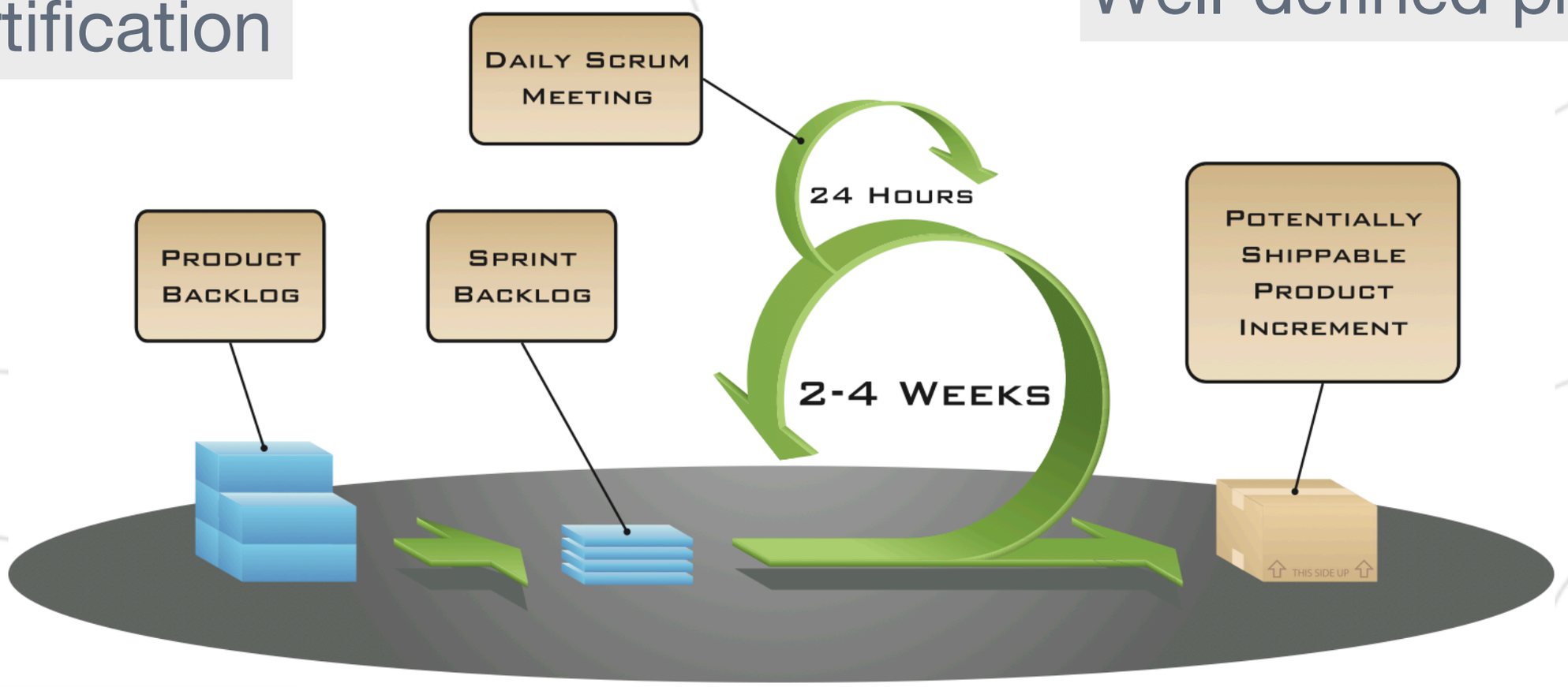
geecon 2013

The „S” word

Long iterations

Well-defined process

Certification



Conferences

SCRUM = Agile

Books

geecon 2013

Coaching


arena
vodafone
Anna

There are certain things you must do...

You must write tests before code.

You must program in pairs.

You must integrate frequently.

You must be rested.

You must communicate with the customer daily.

You must follow the customer's priorities.

You must leave the software clean and simple by the end of the day.

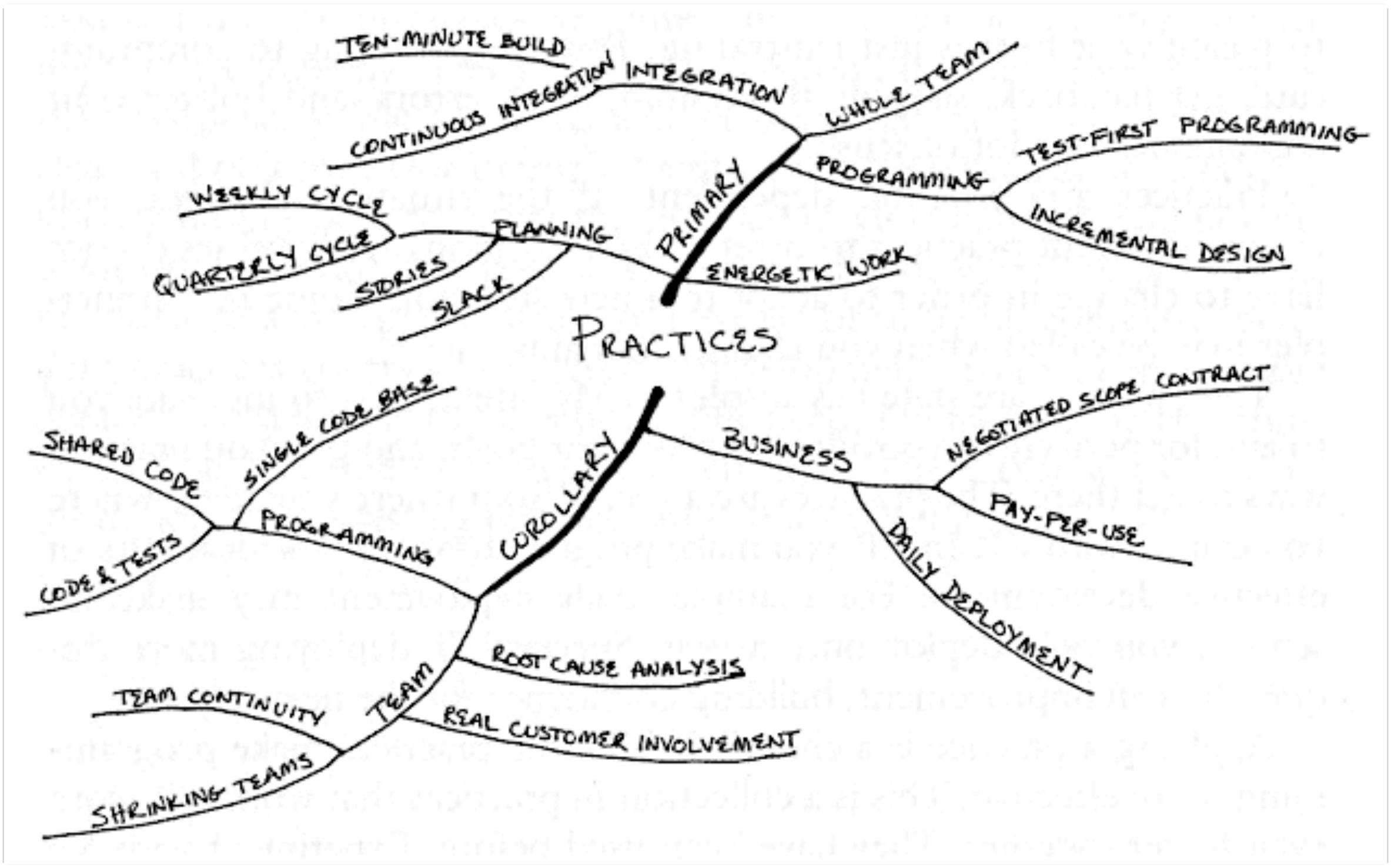
You must adapt the process and practices to your environment.

Why is it called Extreme?

If something is **hard**, let's do it more often.

If something is **good**, let's do it more often.

geecon 2013



Beck, K. (1999). *Extreme Programming Explained: Embrace Change*. Addison-Wesley. ISBN 978-0-321-27865-4.

Ariane 5 flight 501

10 years of work

5.000.000.000 €

geecon 2013

http://www.dutchspace.nl/uploadedImages/Products_and_Services/Launchers/Ariane%205%20Launch%20512%20-%20ESA.JPG

http://www.capcomespace.net/dossiers/espace_europeen/ariane/ariane5/AR501/V88%20explosion%2003.jpg

Oops - an error has occurred - Mozilla Firefox

File Edit View History Bookmarks Tools Help

jira.com https://dev1.jira.com/secure/admin/StudioImport!showUrls.jspa

Google

Diigo Bookmark Highlight Comment Send Message (0) Read Later Unread Recent Add a filter Options Capture

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

[#JST-24... [#JST-28... [#JST-24... Studio sy... Oops... How to b... Apache F... [#STRM-... Maven R... Index of /... Sonatype... Index of /... CR-JST-2... Uploaded... Atlassian... JIRA Studi... Disabling... Maven D...

Oops - an error has occurred

System Error

A system error has occurred.

Please try submitting this problem via the Support Request Page
Otherwise, please create a support issue on our **support system** at http://support.atlassian.com with the following information:

1. a description of your problem
2. cut & paste the error and system information found below
3. attach the application server log file (/data/jirastudio/jira/log/atlassian-jira.log)

Cause:
java.lang.NullPointerException

Stack Trace: [hide]

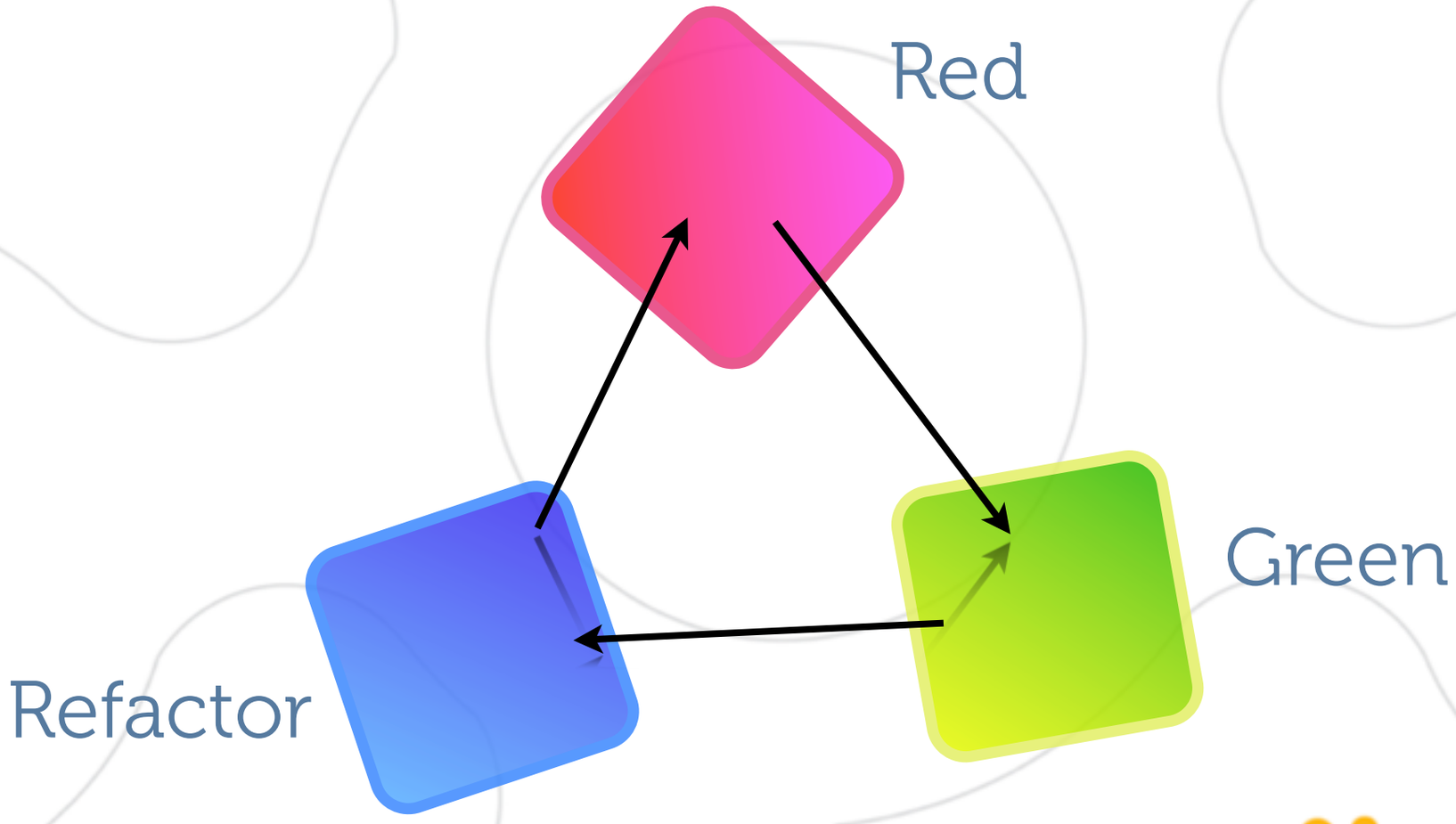
```
java.lang.NullPointerException
        at com.atlassian.jira.studio.importer.StudioImport.doShowUrls(StudioImport.java:52)
        at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
        at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
        at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
        at java.lang.reflect.Method.invoke(Method.java:597)
        at webwork.util.InjectionUtils$DefaultInjectionImpl.invoke(InjectionUtils.java:70)
        at webwork.util.InjectionUtils.invoke(InjectionUtils.java:56)
        at webwork.action.ActionSupport.invokeCommand(ActionSupport.java:433)
        at webwork.action.ActionSupport.execute(ActionSupport.java:157)
        at com.atlassian.jira.action.JiraActionSupport.execute(JiraActionSupport.java:53)
        at com.atlassian.jira.studio.importer.StudioImport.execute(StudioImport.java:42)
        at webwork.dispatcher.GenericDispatcher.executeAction(GenericDispatcher.java:139)
        at com.atlassian.jira.web.dispatcher.JiraServletDispatcher.service(JiraServletDispatcher.java:171)
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:803)
        at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:269)
        at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:188)
        at com.atlassian.core.filters.HeaderSanitisingFilter.doFilter(HeaderSanitisingFilter.java:44)
        at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:215)
        at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:188)
        at com.atlassian.plugin.servlet.filter.IteratingFilterChain.doFilter(IteratingFilterChain.java:46)
        at com.atlassian.plugin.servlet.filter.ServletFilterModuleContainerFilter.doFilter(ServletFilterModuleContainerFilter.java:55)
        at com.atlassian.plugin.servlet.filter.ServletFilterModuleContainerFilter.doFilter(ServletFilterModuleContainerFilter.java:41)
        at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:215)
        at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:188)
        at com.atlassian.jira.web.filters.accesslog.AccessLogFilter.executeRequest(AccessLogFilter.java:99)
        at com.atlassian.jira.web.filters.accesslog.AccessLogFilter.doFilter(AccessLogFilter.java:83)
        at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:215)
        at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:188)
        at com.atlassian.jira.security.xsrf.XsrfTokenAdditionRequestFilter.doFilter(XsrfTokenAdditionRequestFilter.java:50)
        at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:215)
        at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:188)
        at com.opensymphony.module.sitemesh.filter.PageFilter.parsePage(PageFilter.java:119)
        at com.opensymphony.module.sitemesh.filter.PageFilter.doFilter(PageFilter.java:55)
```

Done

Oops - an er... OSGi Service... STAC - fabia... Kwalletma... Liferea 09:49 am Tue, 31 Aug

TDD

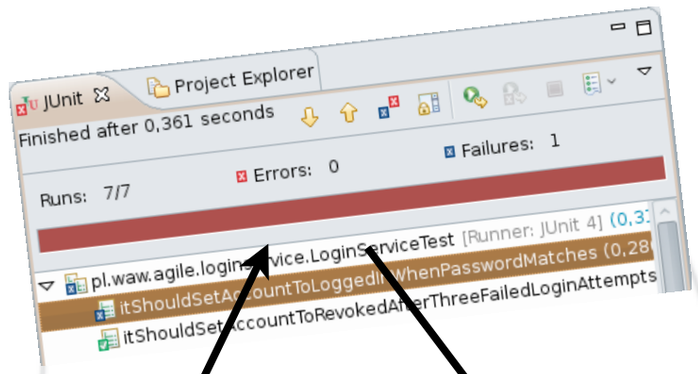
A technique of software development based on repeating a short cycle:



geecon 2013

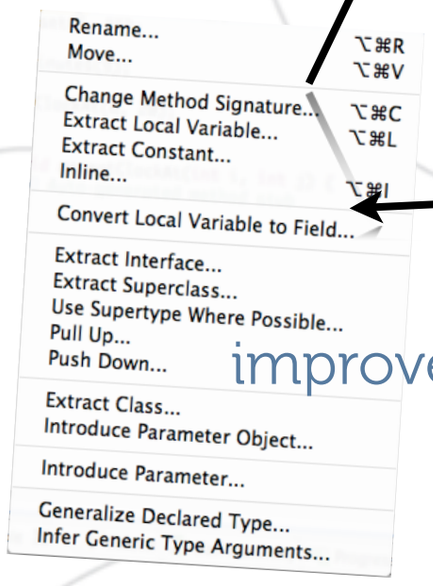
TDD

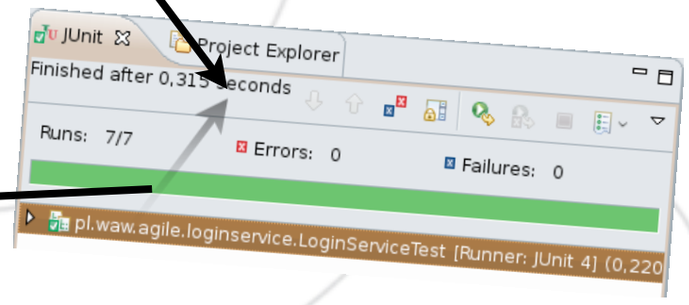
A technique of software development based on repeating a short cycle:



write new, unpassing test

make the test pass

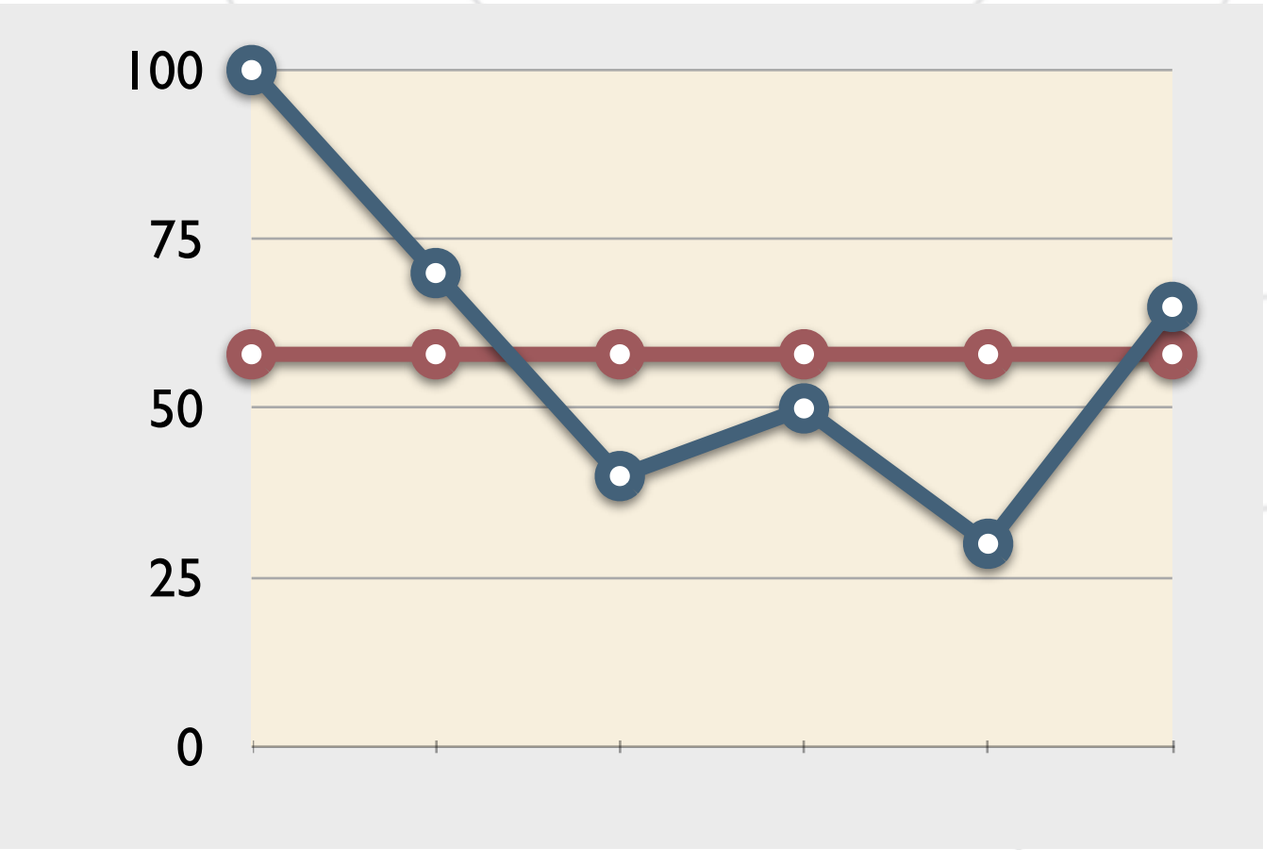




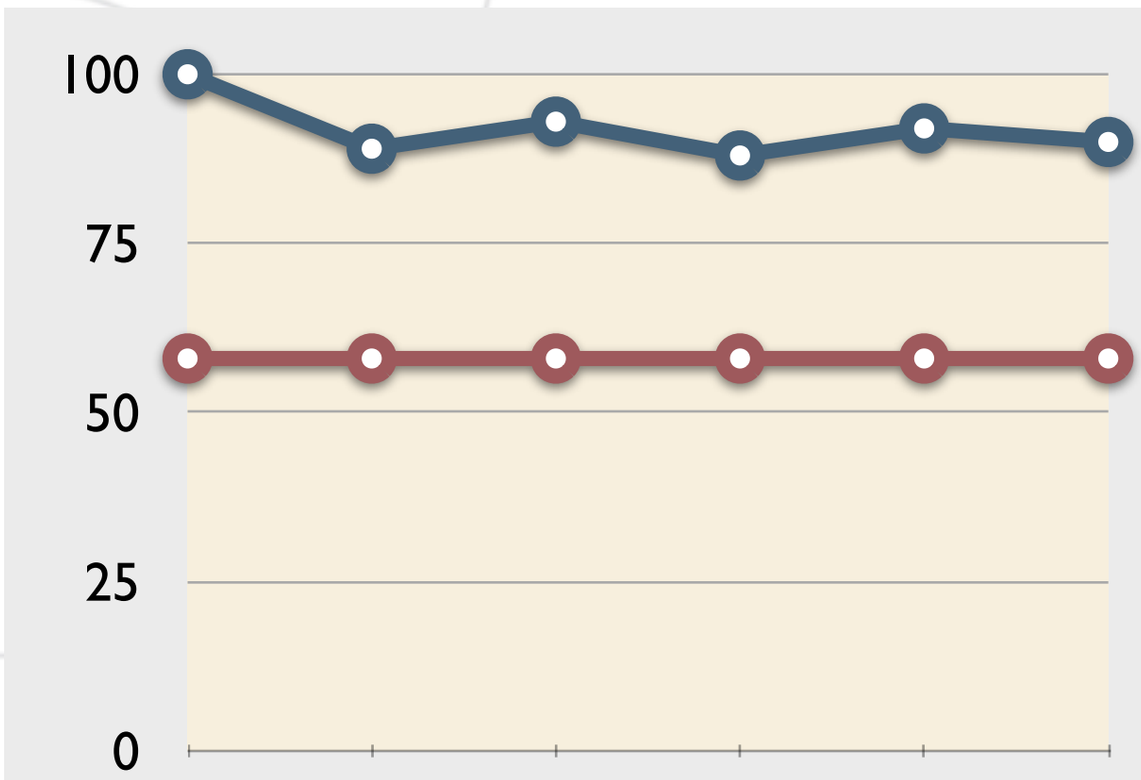
improve the design and code readability

geecon 2013

Acceptable quality



Traditional development



Agile

geecon 2013

http://flagstaffclimbing.com/wp-content/themes/climbing/images/flag-climbing-bg.jpg

Define behaviour
Specify contract

```
Clock clock = new Clock();
clock.set(6, 15);

wait(5).minutes();

assertClockAt(6, 20);
```

Red

- thinking - which way to choose?
- designing

```
Clock clock = new Clock();
clock.set(6, 15);

wait(5).minutes();

assertClockAt(6, 20);
```

- code quality
- testability
- test quality

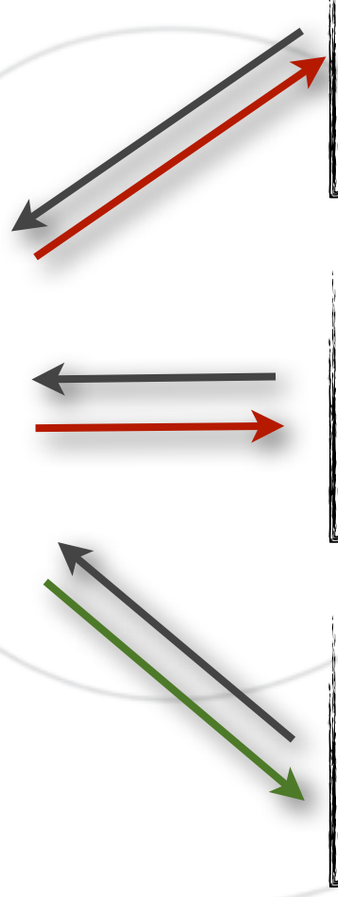
Red ➔ Green

How to achieve the behaviour?

```
Clock clock = new Clock();
clock.set(6, 15);

wait(5).minutes();

assertClockAt(6, 20);
```



```
class Clock {
   ...
}
```

```
class Clock {
   ...
}
```

```
class Clock {
   ...
}
```

geecon 2013

Red ➔ Green

programming

```
Clock clock = new Clock();
clock.set(6, 15);

wait.minutes(5);

assertClockAt(6, 20);
```

```
class Clock {
   ...
}
```

```
class Clock {
   ...
}
```

```
class Clock {
   ...
}
```

- is the new code working?
- didn't I break any other piece of code?
- did I choose the right test size? (coding time)

Refactor

How should the code look like?

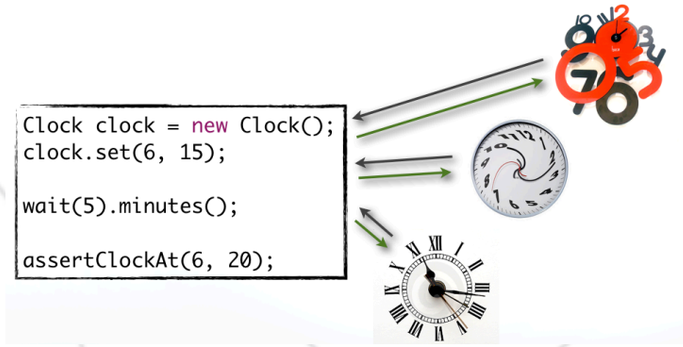
```
Clock clock = new Clock();
clock.set(6, 15);

wait(5).minutes();

assertClockAt(6, 20);
```

Refactor

- designing
- refactoring



- design quality (how easy it is to refactor)
- tests' quality (how safe is it to refactor)

How to start doing it?

JUnit junitparams arquilian

hamcrest mockito jMock

fest-assert runners TestNG

catch-exception surefire PowerMock

How to start doing it?

Just do it.

geecon 2013

Ok, but... how to make it stick?

Do it with the whole team.

Get a coach to spend time with your team, on your code.

Learn it in pairs

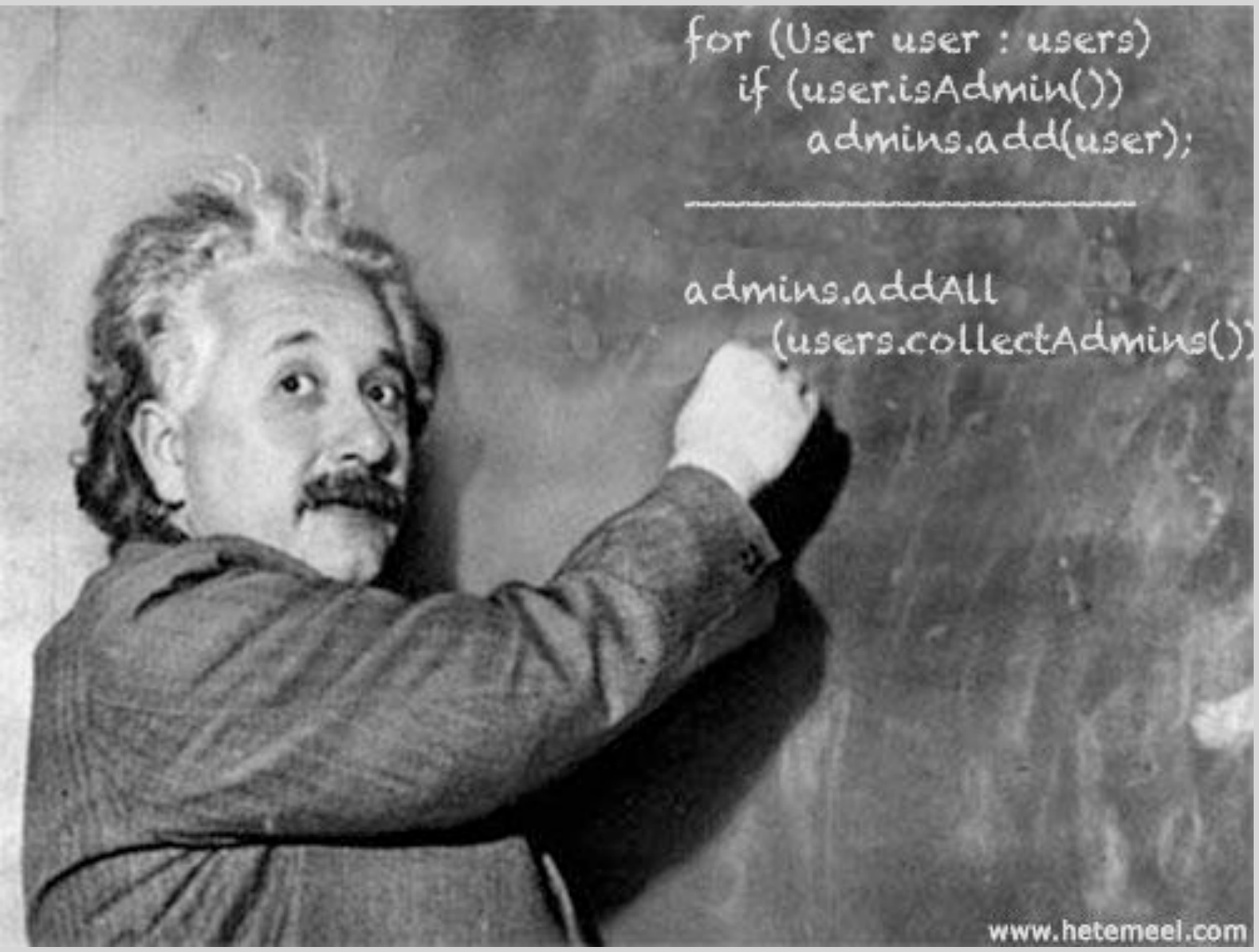
Try kata trainings - it will build a habit of test-driving in your head.

Peer-review your test code

Refactoring

> Any fool can write code that a computer can understand.
> Good programmers write code that humans can understand.

Martin Fowler



geecon 2013

```
public int[] returnPoints (Theme usersTheme) {
        int[] points = new int[10];
        if (year != 0 && year >= usersTheme.year - 15 && year <= usersTheme.year + 15) {
            if (year >= usersTheme.year - 10 && year <= usersTheme.year + 10) {
                if (year >= usersTheme.year - 5 && year <= usersTheme.year + 5) {
                    if (year >= usersTheme.year - 2 && year <= usersTheme.year + 2) {
                        points[0] = 4;
                    } else
                        points[0] = 3;
                } else
                    points[0] = 2;
            } else
                points[0] = 1;
        }
        if (composer == usersTheme.composer) {
            points[1] = 8;
        }
        if (category != 2) {
            if (_11 != 0 && _11 == usersTheme._11)
                points[4] = 4;
            if (_12 != 0 && _12 == usersTheme._12)
                points[5] = 3;
            if (_13 != 0 && _13 == usersTheme._13)
                points[6] = 2;
            if (_15 != 0 && _15 == usersTheme._15)
                points[8] = 2;
            if (_16 != 0 && _16 == usersTheme._16)
                points[9] = 4;
            if (category == 1) {
                if (((_9 == 1 || _9 == 3) && _9 == usersTheme._9) || ((_9 == 2 || _9 == 4) && _9 == usersTheme._9 && _10 == usersTheme._10))
                    points[2] = 6;
            } else {
                if (_9 == usersTheme._9 && _10 == usersTheme._10)
                    points[2] = 6;
            }
        } else {
            if (_9 != 0 && _9 == usersTheme._9)
                points[2] = 6;
            if (_10 != 0 && _10 == usersTheme._10)
                points[3] = 3;
            if (Application.getConfiguration().getQuestions() == Configuration.QUESTIONS_FULL_SET) {
                if ((_11 != 0 && _11 == usersTheme._11 && _13 != 0 && _13 == usersTheme._13)
                        || (_11 != 0 && _11 == usersTheme._12 && _13 != 0 && _13 == usersTheme._14))
                    points[4] = 4;
                if ((_12 != 0 && _12 == usersTheme._12 && _14 != 0 && _14 == usersTheme._14)
                        || (_12 != 0 && _12 == usersTheme._11 && _14 != 0 && _14 == usersTheme._13))
                    points[6] = 4;
            } else {
                if (_13 != 0 && (_13 == usersTheme._13 || _13 == usersTheme._14))
                    points[4] = 4;
                if (_14 != 0 && (_14 == usersTheme._14 || _14 == usersTheme._13))
                    points[6] = 4;
            }
            if (_15 != 0 && _15 == usersTheme._15)
                points[8] = 4;
            if (_16 != 0 && _16 == usersTheme._16)
                points[9] = 2;
        }
        return points;
    }
```

// https://code.google.com/p/gag/

@LOL @Facepalm

@WTF("who's gonna refactor this one?")

Refactoring

- no change in code semantics
- improves readability
- improves flexibility
- improves performance

Ok, but how to start doing it?

Have automated tests!

Learn a bit about OOD...

Design Patterns

Baby steps - as a part of your TDD cycle at best.

S.O.L.I.D.

Check what your IDE can do automatically

...and master it :-)

And how to make it stick?

Master your IDE

Become an OO master

Big refactorings only when
absolutely needed.
Don’t do it just for art’s sake.

Force yourself to think what to refactor
in every TDD cycle.

Pair programming

http://static.guim.co.uk/sys-images/Guardian/Pix/pictures/2011/10/25/13195652461 30/Russian-President-Dmitry--007.jpg

The biggest challenge for me personally was essentially mourning for the death of “Programmer Man”.

Programmer Man is how I think of myself when I’ve got my headphones in, speed metal blaring in my ears, and I’m coding like a motherfucker. My fingers can’t keep up with my brain. I’m In The Zone.

For most of my career, this image is what I’ve considered to be the zenith. When I come home and was in Programmer Man Mode most of the day, I feel like I’ve had a good day.

Pair Programming undeniably killed Programmer Man. This was a tough adjustment, since I’ve considered that mode to be my favorite for so long. I now see, however, that **Programmer Man was, without me knowing it, Technical Debt Man.**

http://www.nomachetejuggling.com/2009/02/21/i-love-pair-programming/

Don’t be afraid of pair-programming - you’re not as good as you think, but you’re not as bad as you fear.

Ron Jeffries

Ok, but how to start doing it?

Comfortable position

Do harder bits in pairs first

Get a shower.

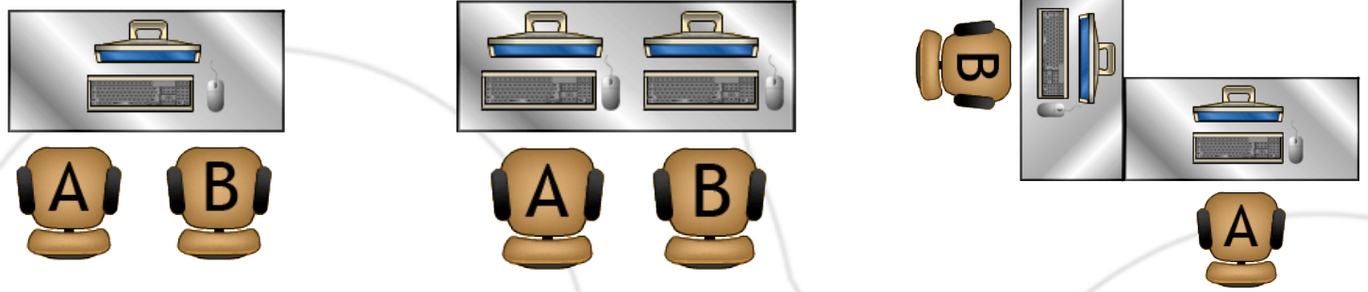
Do it the right way - one person codes,
the other gets the bigger picture.

Switch pairs.

And how to make it stick?

2 keyboards

2 mice



Check which setting fits you best.

http://www.nomachetejuggling.com/2011/08/25/mechanics-of-good-pairing/

Initially everybody MUST pair. Make it optional once you know it well.

geecon 2013

Collective code ownership

John, would you be so nice...

Who wrote it?

Whose bug is it?

WTF???

Who is the owner of...

Hey, whose is that class?

Sorry, I've got other things to do now.

Mary, you're responsible for it, ain't ya?

We cannot fix it - Steve's ill.

Ok, but how to start doing it?

Choose one module to own collectively

Have automated tests!

Discuss what you’re planning to work on.

„Cannot refuse helping” rule.

Ask for help.

Pair program as much as you can.

Make technical presentations / discussions

Choose tasks you have little clue about.

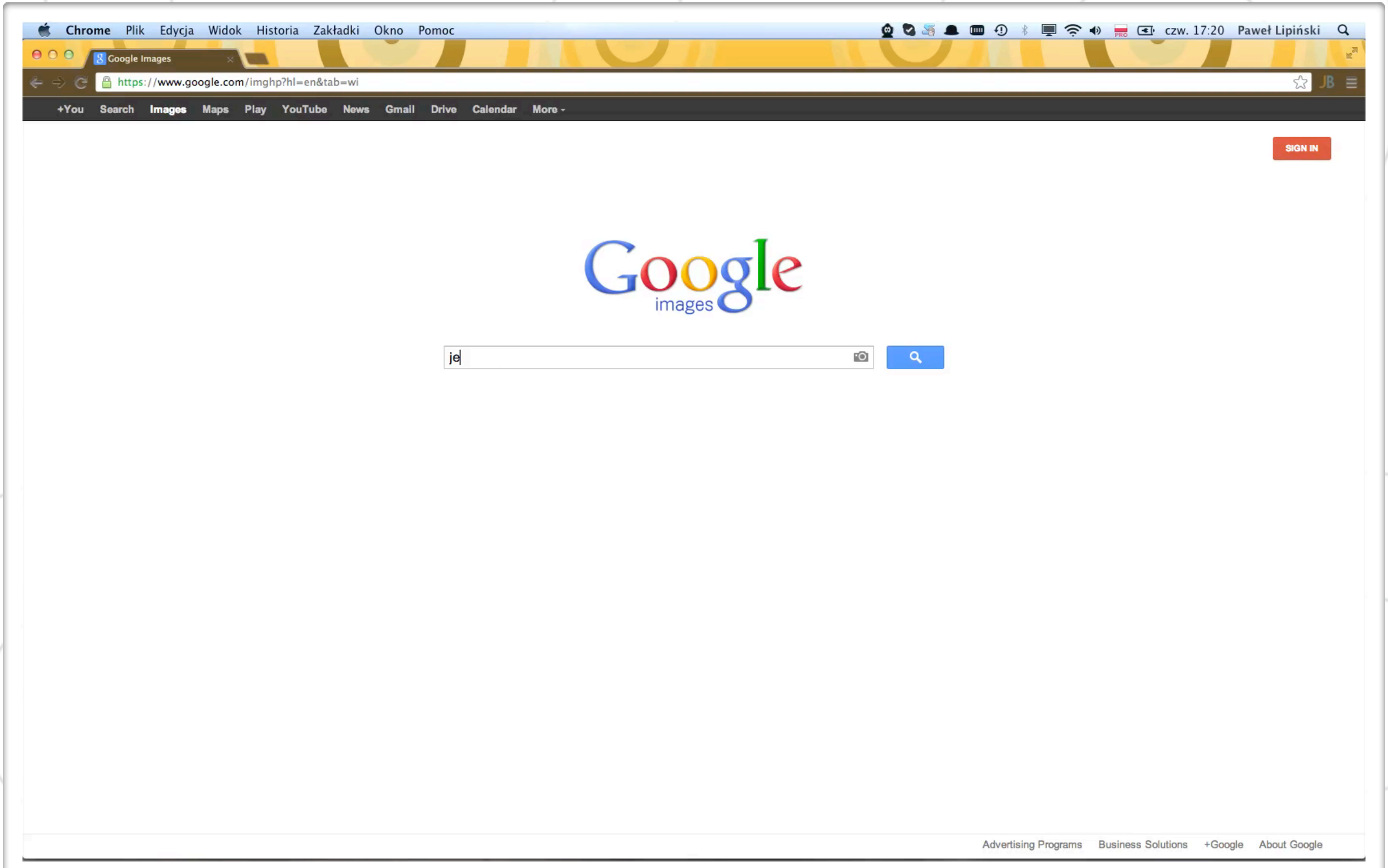
And how to make it stick?

It’s about team culture - discuss it.

Pair program.

Peer-review everything that hasn’t been pair-programmed.

Initially everybody MUST pair. Make it optional once you know it well.


Chrome Plik Edycja Widok Historia Zakładki Okno Pomoc
czw. 17:20 Paweł Lipiński
Google Images
https://www.google.com/imghp?hl=en&tab=wi
+You Search Images Maps Play YouTube News Gmail Drive Calendar More
SIGN IN
Google
images
je
Advertising Programs Business Solutions +Google About Google

Ok, but how to start doing it?

Check-in often... clean builds.

Never leave the office if the build is broken.

Don’t comment out failing tests...

Jenkins

cruise-control

gump

Team Foundation Server

TeamCity

Continuum

Hudson

Bamboo

Time-box fixing build revert if needed.

You’re responsible if your build broke something.

And how to make it stick?

Optimize your building process.

Sonar

Lots of tests.

Keep your tests fast.

Integration tests.

End-to-end tests.

Notifications which piss you off.

Fancy info radiator.

geecon 2013

Coding Standards

Refactoring

Continuous Integration

40-hour week

Pair Programming

Simple Design

Collective Code Ownership

TDD

Metaphor

geecon 2013

Diversity

Team

Growth

Fun

Effectiveness

Extreme Programming practices

Business-driven

Motivation

Efficiency

Skills

Quality

geecon 2013

let's **move**
the **java world**

Thank you!

pawel.lipinski@pragmatists.pl