

Du `println()` aux tests automatisés et à l'intégration continue techniques et outils

Mathieu Acher

Maître de Conférences

@acherm

mathieu.acher@irisa.fr

Material

<https://github.com/acherm/PDL1920>

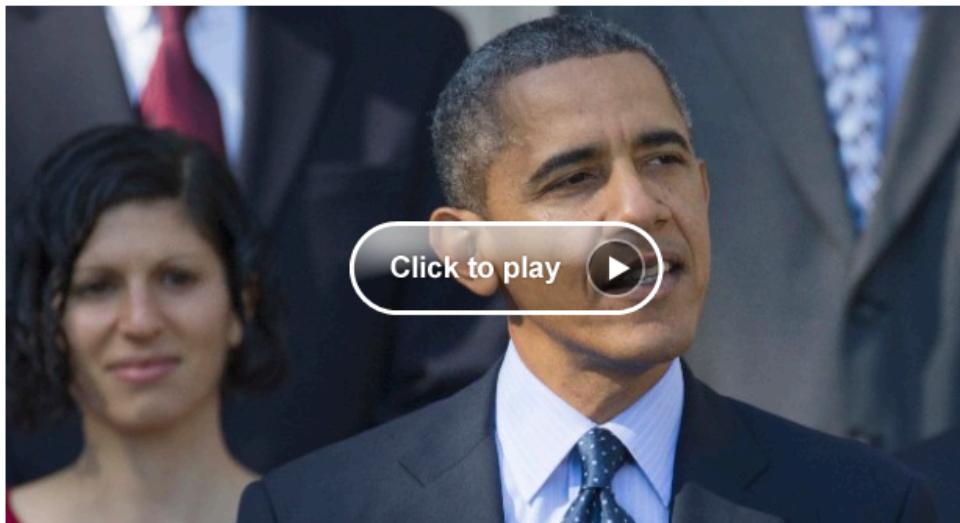
Travail actuel

- Compréhension du projet eg
 - <http://blog.mathieuacher.com/WikipediaMatrixChallenge/>
- Audit d'un projet existant
 - documentation (README.md + DESIGN.md), Github issues, INSTALL.md (donc possiblement fixer la procédure d'installation)
- Installation (ce qu'il est attendu!)
 - Tests automatiques
 - Avec Maven (indépendamment d'Eclipse/IntelliJ/VScode/...)
 - Intégration continue

Report: Healthcare website failed test ahead of rollout

By Ed Payne, Matt Smith and Tom Cohen, CNN

October 23, 2013 -- Updated 0103 GMT (0903 HKT)



Report: Obamacare site failed early test

STORY HIGHLIGHTS

- **NEW:** Top White House official part of "tech surge" on Obamacare
- Obamacare "is not failing" despite website woes, White House spokesman says
- Obama says HealthCare.gov problems are "going to get fixed"
- Secretary Sebelius expected to testify at a congressional hearing next week

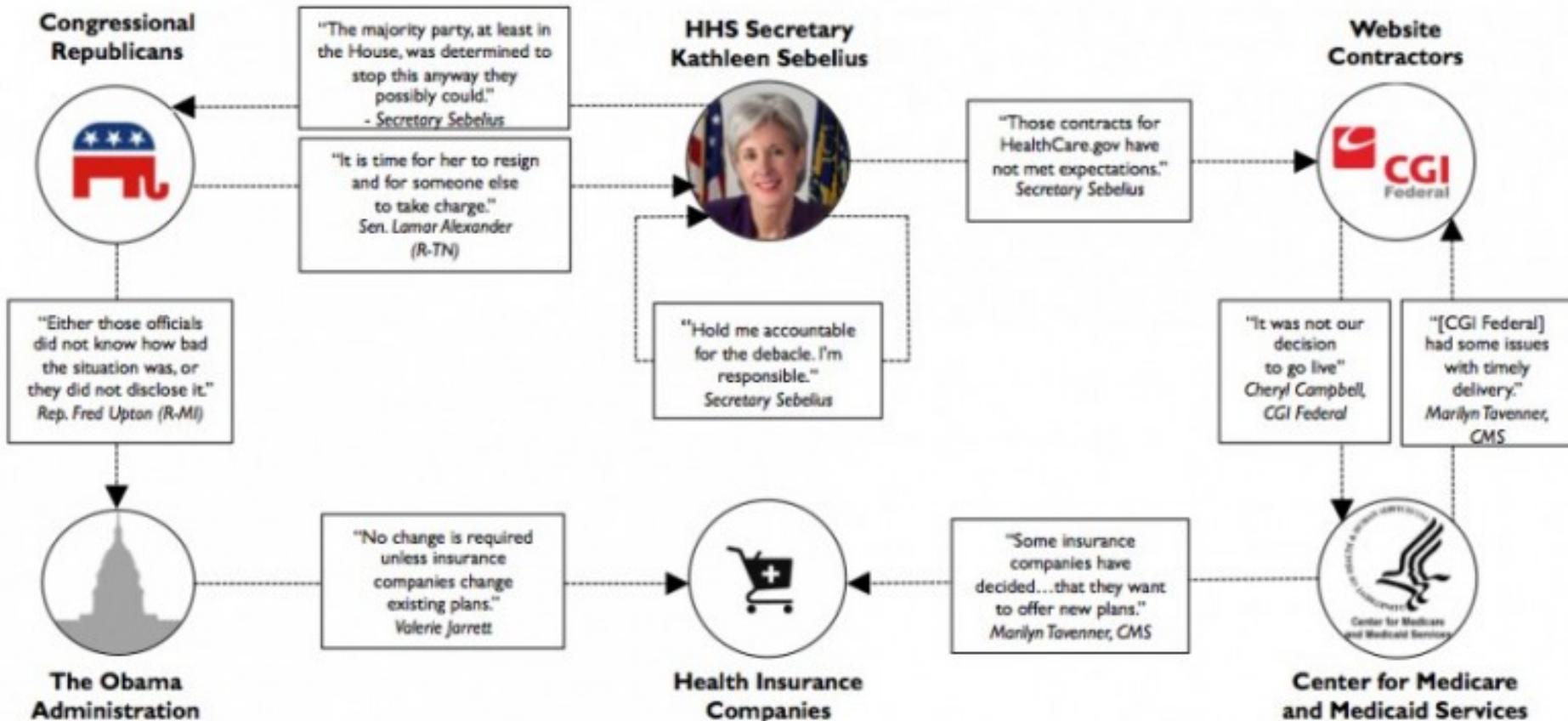
Washington (CNN) -- The President's healthcare sign-up web page was supposed to handle tens of thousands of people at once. But in a trial run days before its launch, just a few hundred users flatlined the site.

Despite the problems, federal health officials pushed aside the crash cart and rolled out HealthCare.gov on October 1 as planned, [The Washington Post reported](#).

The result? The website crashed shortly after midnight as a couple thousand people tried to start the process, two people familiar with the project told the Post.

Requirements engineering/ Management problem

ACA Finger-Pointing Flowchart



<http://www.washingtonpost.com/blogs/wonkblog/wp/2013/11/01/thirty-one-things-we-learned-in-healthcare-govs-first-31-days/>

Thirty-one things we learned in HealthCare.gov's first 31 days

Scalability problem

**Technical problems (e.g.,
inaccurate data, cancellation
failures)**

Testing issues

<http://www.washingtonpost.com/blogs/wonkblog/wp/2013/11/01/thirty-one-things-we-learned-in-healthcare-govs-first-31-days/>

10. HealthCare.gov didn't have enough testing before going live.

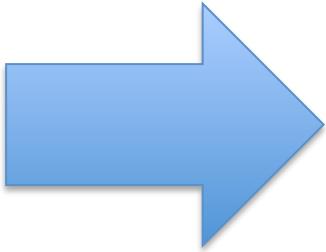
This became clear in a series of Congressional hearings, where federal contractors testified that end-to-end testing only began in the final weeks of September, right before the Oct. 1 launch. When pressed on how much time would have been ideal for testing, one contractor told lawmakers that “months would have been nice.”

<http://www.washingtonpost.com/blogs/wonkblog/wp/2013/11/01/thirty-one-things-we-learned-in-healthcare-govs-first-31-days/>

System.out.println()



WIKIPEDIA
The Free Encyclopedia



CSV (Comma Separated Values)

Product	Image process...	Sensor format	Sensor type	Sensor manufac...	Megapixels	Focus points	Metering pixels	Viewfinder cov...
D3X	EXPPEED	Full-frame	CMOS	Sony	24.5	51	1005	100%
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D1X	-	APS-C	CCD	Sony	5.3	5	1005	96%
D1	-	APS-C	CCD	Sony	2.66	5	1005	96%
D4S	EXPPEED 4	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D4	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D3S	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D3	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D2Hs	-	APS-C	JETEL-BCAST	Nikon	4.1	11	1005	100%
D2H	-	APS-C	JETEL-BCAST	Nikon	4.1	11	1005	100%
D1H	-	APS-C	CCD	Sony	2.7	5	1005	96%
D810	EXPPEED 4	Full-frame	CMOS	Sony	36.5	51	91000	100%
D600	EXPPEED 3	Full-frame	CMOS	Sony	36.3	51	91000	100%
D700	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	96%
D700	EXPPEED 4	Full-frame	CMOS	Nikon	24.9	51	91000	100%
Df	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%

Comparison of Canon EOS digital cameras

From Wikipedia, the free encyclopedia

The following tables provide general information as well as a comparison of technical specifications for a number of [Canon EOS digital cameras](#).

General information [\[edit\]](#)

Model	Image processor	Sensor format	Megapixels	Min ISO	Max ISO	Autofocus points	Viewfinder magnification, coverage	Display size, dots (ratio)	Touchscreen	Live view	Max FPS	Storage	Release date	Weight (kg)	Dimensions, WxHxD (mm)	Video	Main battery
1Ds	DIGIC	Full-frame CMOS	11.4	100	1250	45	0.70x, 100%	2.0", 120k	No	No	3	CF	2002Q4	1.265	156 × 158 × 80	-	NP-E3
1Ds Mk II	DIGIC II	Full-frame CMOS	16.7	50	3200	45	0.70x, 100%	2.0", 230k	No	No	4.5	CF, SD	2004Q4	1.215	156 × 158 × 80	-	NP-E3
1Ds Mk III	Dual DIGIC III	Full-frame CMOS	21.1	50	3200	45	0.76x, 100%	3.0", 230k	No	Yes	5.0	CF, SD	2007Q4	1.210	156 × 160 × 80	-	LP-E4
1D	DIGIC	APS-H CCD	4	100	3200	45	0.72x, 100%	2.0", 120k	No	No	8.0	CF	2001Q4	1.250	156 × 158 × 80	-	NP-E3
1D Mk II	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72x, 100%	2.0", 230k	No	No	8.5	CF, SD	2004Q2	1.220	156 × 158 × 80	-	NP-E3
1D Mk II N	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72x, 100%	2.5", 230k	No	No	8.5	CF, SD	2005Q3	1.225	156 × 158 × 80	-	NP-E3
1D Mk III	Dual DIGIC III	APS-H CMOS	10.1	50	6400	45	0.76x, 100%	3.0", 230k	No	Yes	10	CF, SD	2007Q1	1.155	156 × 157 × 80	-	LP-E4
1D Mk IV	Dual DIGIC 4	APS-H CMOS	16.1	50	102,400	45	0.76x, 100%	3.0", 920k	No	Yes	10	CF, SD	2009Q4	1.180	156 × 157 × 80	1080p30	LP-E4
1D X	Dual DIGIC 5	Full-frame CMOS	18.1	50	204,800	61	0.76x, 100%	3.2", 1040k	No	Yes	14	CF (x2)	2012Q2	1.530	158 × 164 × 83	1080p30, 720p60	LP-E4N

Imaginons: l'exécution de l'extracteur Wikitext aboutit à un RuntimeException sur cette page spécifiquement (via exécution d'un programme Java)

Problème: localiser et comprendre l'erreur!

Comparison of Canon EOS digital cameras

From Wikipedia, the free encyclopedia

The following tables provide general information as well as a comparison of technical specifications for a number of [Canon EOS digital cameras](#).

General information [\[edit\]](#)

Model	Image processor	Sensor format	Megapixels	Min ISO	Max ISO	Autofocus points	Viewfinder magnification, coverage	Display size, dots (ratio)	Touchscreen	Live view	Max FPS	Storage	Release date	Weight (kg)	Dimensions, WxHxD (mm)	Video	Main battery
1Ds	DIGIC	Full-frame CMOS	11.4	100	1250	45	0.70x, 100%	2.0", 120k	No	No	3	CF	2002Q4	1.265	156 × 158 × 80	-	NP-E3
1Ds Mk II	DIGIC II	Full-frame CMOS	16.7	50	3200	45	0.70x, 100%	2.0", 230k	No	No	4.5	CF, SD	2004Q4	1.215	156 × 158 × 80	-	NP-E3
1Ds Mk III	Dual DIGIC III	Full-frame CMOS	21.1	50	3200	45	0.76x, 100%	3.0", 230k	No	Yes	5.0	CF, SD	2007Q4	1.210	156 × 160 × 80	-	LP-E4
1D	DIGIC	APS-H CCD	4	100	3200	45	0.72x, 100%	2.0", 120k	No	No	8.0	CF	2001Q4	1.250	156 × 158 × 80	-	NP-E3
1D Mk II	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72x, 100%	2.0", 230k	No	No	8.5	CF, SD	2004Q2	1.220	156 × 158 × 80	-	NP-E3
1D Mk II N	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72x, 100%	2.5", 230k	No	No	8.5	CF, SD	2005Q3	1.225	156 × 158 × 80	-	NP-E3
1D Mk III	Dual DIGIC III	APS-H CMOS	10.1	50	6400	45	0.76x, 100%	3.0", 230k	No	Yes	10	CF, SD	2007Q1	1.155	156 × 157 × 80	-	LP-E4
1D Mk IV	Dual DIGIC 4	APS-H CMOS	16.1	50	102,400	45	0.76x, 100%	3.0", 920k	No	Yes	10	CF, SD	2009Q4	1.180	156 × 157 × 80	1080p30	LP-E4
1D X	Dual DIGIC 5	Full-frame CMOS	18.1	50	204,800	61	0.76x, 100%	3.2", 1040k	No	Yes	14	CF (x2)	2012Q2	1.530	158 × 164 × 83	1080p30, 720p60	LP-E4N

Imaginons: l'exécution de l'extracteur Wikitext aboutit à un fichier vide sur cette page spécifiquement (via exécution d'un programme Java)

Problème: localiser et comprendre l'erreur!

Comparison of Canon EOS digital cameras

From Wikipedia, the free encyclopedia

The following tables provide general information as well as a comparison of technical specifications for a number of [Canon EOS digital cameras](#).

General information [\[edit\]](#)

Model	Image processor	Sensor format	Megapixels	Min ISO	Max ISO	Autofocus points	Viewfinder magnification, coverage	Display size, dots (ratio)	Touchscreen	Live view	Max FPS	Storage	Release date	Weight (kg)	Dimensions, WxHxD (mm)	Video	Main battery
1Ds	DIGIC	Full-frame CMOS	14.1	100	1250	45	0.70x, 100%	2.0", 120k	No	No	3	CF	2001Q4	1.265	156 x 158 x 80	-	NP-E3
1Ds Mk II	DIGIC II	Full-frame CMOS	16.7	50	3200	45	0.70x, 100%	2.0", 230k	No	No	4	CF, SD	2004Q4	1.215	156 x 158 x 80	-	NP-E3
1Ds Mk III	Dual DIGIC III	Full-frame CMOS	21.1	50	3200	45	0.76x, 100%	3.0", 230k	No	Yes	5.0	CF, SD	2007Q4	1.210	156 x 160 x 80	-	LP-E4
1D	DIGIC	APS-H CCD	4	100	3200	45	0.72x, 100%	2.0", 120k	No	No	8.0	CF	2001Q4	1.250	156 x 158 x 80	-	NP-E3
1D Mk II	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72x, 100%	2.0", 230k	No	No	8.5	CF, SD	2004Q2	1.220	156 x 158 x 80	-	NP-E3
1D Mk II N	DIGIC II	APS-H CMOS	8.2	50	3200	45	0.72x, 100%	2.5", 230k	No	No	8.5	CF, SD	2005Q3	1.225	156 x 158 x 80	-	NP-E3
1D Mk III	Dual DIGIC III	APS-H CMOS	10.1	50	6400	45	0.76x, 100%	3.0", 230k	No	Yes	10	CF, SD	2007Q1	1.155	156 x 157 x 80	-	LP-E4
1D Mk IV	Dual DIGIC 4	APS-H CMOS	16.1	50	102,400	45	0.76x, 100%	3.0", 920k	No	Yes	10	CF, SD	2009Q4	1.180	156 x 157 x 80	1080p30	LP-E4
1DX	Dual DIGIC 5	Full-frame CMOS	18.1	50	204,800	61	0.76x, 100%	3.2", 1040k	No	Yes	14	CF (x2)	2012Q2	1.530	158 x 164 x 83	1080p30, 720p60	LP-E4N

```
public void ToCSV() throws IOException {
    String result = "";
    FormatWikitext clone = clone();
    String[] separateur = clone.wikitext.split("wikitable");
    int nbtab = separateur.length -1;
    String title = getTitle();
    for(int i = 0; i< nbtab; i++){
        tabCourant = i +1;
        ProductionCSV head = headToCSV();
        ProductionCSV body = rowToCSV();
        result = (head.csv + "\n" +body.csv);
        ProductionCSV prod = new ProductionCSV(result);
        prod.generateCSVFromWikitext(title, tabCourant);
        //System.out.println(i+1 + " tableaux traitÃ©s");
    }
}
```

logging (println)?

Logging, why? (claims)

- Logging is easier than debugging
- Logging is faster than debugging
- Logging can work in environments where debugging is not supported
- Can work in production environments
- Logs can be referenced anytime in future as the data is stored



Logging



- Logging is chronological and systematic record of data processing events in a program
 - e.g. the Windows Event Log
- Logs can be saved to a persistent medium to be studied at a later time
- Use logging in the development phase:
 - Logging can help you **debug** the code
- Use logging in the production environment:
 - Helps you **troubleshoot problems**

Logging Methods, How?

- The evil `System.out.println()`
- Custom Solution to Log to various datastores,
eg text files, db, etc...
- Use Standard APIs
 - Don't reinvent the wheel

Log4J



- Popular logging frameworks for Java
- Designed to be reliable, fast and extensible
- Simple to understand and to use API
- Allows the developer to control which log statements are output with arbitrary granularity
- Fully configurable at runtime using external configuration files

Log4J Architecture



- Log4J has three main components: loggers, appenders and layouts
 - **Loggers**
 - Channels for printing logging information
 - **Appenders**
 - Output destinations (console, File, Database, Email/SMS Notifications, Log to a socket, and many others...)
 - **Layouts**
 - Formats that appenders use to write their output
- **Priorities**

Logger

- Responsible for Logging
- Accessed through java code
- Configured Externally
- Every Logger has a name
- Prioritize messages based on level
 - TRACE, DEBUG, INFO, WARN, ERROR & FATAL
- Usually named following dot convention like java classes do.
 - Eg com.foo.bar.ClassName
- Follows inheritance based on name

Logger API

- **Factory methods to get Logger**

- `Logger.getLogger(Class c)`
 - `Logger.getLogger(String s)`

- **Method used to log message**

- `trace()`, `debug()`, `info()`, `warn()`, `error()`, `fatal()`
 - Details
 - `void debug(java.lang.Object message)`
 - `void debug(java.lang.Object message, java.lang.Throwable t)`
 - Generic Log method
 - `void log(Priority priority, java.lang.Object message)`
 - `void log(Priority priority,
 java.lang.Object message, java.lang.Throwable t)`

Root Logger

- The root logger resides at the top of the logger hierarchy. It is exceptional in two ways:
 1. it always exists,
 2. it cannot be retrieved by name.
- `Logger.getRootLogger()`

Appender

- Appenders put the log messages to their actual destinations.
- No programmatic change is required to configure appenders
- Can add multiple appenders to a Logger.
- Each appender has its Layout.
- ConsoleAppender, DailyRollingFileAppender, FileAppender, JDBCAppender, JMSAppender, NTEventLogAppender, RollingFileAppender, SMTPAppender, SocketAppender, SyslogAppender, TelnetAppender

Layout

- Used to customize the format of log output.
- Eg. HTMLLayout, PatternLayout, SimpleLayout, XMLLayout
- Most commonly used is PatternLayout
 - Uses C-like syntax to format.
 - Eg. "%-5p [%t]: %m%n
 - DEBUG [main]: Message 1 WARN [main]: Message 2

Log4j Basics

- Who will log the messages?
 - The Loggers
- What decides the priority of a message?
 - Level
- Where will it be logged?
 - Decided by Appender
- In what format will it be logged?
 - Decided by Layout

Log4j in Action

```
// get a logger instance named "com.foo"
Logger logger = Logger.getLogger("com.foo");

// Now set its level. Normally you do not need to set the
// level of a logger programmatically. This is usually done
// in configuration files.
logger.setLevel(Level.INFO);

Logger barlogger = Logger.getLogger("com.foo.Bar");

// This request is enabled, because WARN >= INFO.
logger.warn("Low fuel level.");

// This request is disabled, because DEBUG < INFO.
logger.debug("Starting search for nearest gas station.");

// The logger instance barlogger, named "com.foo.Bar",
// will inherit its level from the logger named
// "com.foo" Thus, the following request is enabled
// because INFO >= INFO.
barlogger.info("Located nearest gas station.");

// This request is disabled, because DEBUG < INFO.
barlogger.debug("Exiting gas station search");
```

Layouts bis (eg colorizing Logs)

- <http://logging.apache.org/log4j/2.x/manual/layouts.html>
- <http://jeanchristophegay.com/de-la-couleur-dans-les-logs/>

```
<?xml version="1.0" encoding="UTF-8"?>
<configuration status="OFF">
    <appenders>
        <Console name="Console" target="SYSTEM_OUT">
            <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %highlight{%-5level} %logger{36} - %msg%n"/>
        </Console>
    </appenders>
    <loggers>
        <root level="trace">
            <appender-ref ref="Console"/>
        </root>
    </loggers>
</configuration>
```

```
19:54:42.838 [main] TRACE com.github.jcgay.example.log.log4j2.Main - a trace message.
19:54:42.841 [main] DEBUG com.github.jcgay.example.log.log4j2.Main - a debug message.
19:54:42.842 [main] INFO com.github.jcgay.example.log.log4j2.Main - an info message.
19:54:42.842 [main] WARN com.github.jcgay.example.log.log4j2.Main - a warn message.
19:54:42.843 [main] ERROR com.github.jcgay.example.log.log4j2.Main - a error message.
19:54:42.845 [main] FATAL com.github.jcgay.example.log.log4j2.Main - a fatal message.
```

Log4j Optimization & Best Practises

- Use logger as private static variable
- Only one instance per class
- Name logger after class name
- Don't use too many appenders
- Don't use time-consuming conversion patterns
(see javadoc)
- Use Logger.isDebugEnabled() if need be
- Prioritize messages with proper levels



Logging in JavaScript / NodeJS

- Not only in Java!
- Alternatives also in other languages...
- <https://github.com/flatiron/winston>

```
var winston = require('winston');

winston.log('info', 'Hello distributed log files!');
winston.info('Hello again distributed logs');
```

Using Logging Levels

Setting the level for your logging message can be accomplished in one of two ways. You can pass a string representing the logging level to the `log()` method or use the level specified methods defined on every `winston` Logger.

```
//  
// Any Logger instance  
  
logger.log('silly', "127.0.0.1 - there's no place like home");  
logger.log('debug', "127.0.0.1 - there's no place like home");  
logger.log('verbose', "127.0.0.1 - there's no place like home");  
logger.log('info', "127.0.0.1 - there's no place like home");  
logger.log('warn', "127.0.0.1 - there's no place like home");  
logger.log('error', "127.0.0.1 - there's no place like home");  
  
logger.info("127.0.0.1 - there's no place like home");  
logger.warn("127.0.0.1 - there's no place like home");  
logger.error("127.0.0.1 - there's no place like home");
```

Colorization (back)

```
var myCustomLevels = {
  levels: {
    foo: 0,
    bar: 1,
    baz: 2,
    foobar: 3
  },
  colors: {
    foo: 'blue',
    bar: 'green',
    baz: 'yellow',
    foobar: 'red'
  }
};

var customLevelLogger = new (winston.Logger)({ levels: myCustomLevels.levels });
customLevelLogger.foobar('some foobar level-ed message');
```

Debugging

Debug - FAMILIAR/src/main/java/fr/familiar/standalone/FML.java - Eclipse - /Users/macher1/Documents/workspaceFML

Quick Access Java Team Synchronizing SVN Repository Explorer

Debug Package Explorer

FML [Java Application]

fr.familiar.standalone.FML at localhost:51307

Thread [main] (Suspended (breakpoint at line 137 in FML))

 = FML.main(String[]) line: 137

 Daemon Thread [EMF Reference Cleaner] (Running)

 Daemon Thread [com.google.inject.internal.util.\$Finalizer] (Running)

/Library/Java/JavaVirtualMachines/jdk1.7.0_13.jdk/Contents/Home/bin/java (17 nov. 2014 12:58:37)

Variables Breakpoints

Name	Value
args	String[0] (id=19)
jsap	JSAP (id=20)
sw1	Switch (id=24)
sw2	Switch (id=28)
sw3	Switch (id=29)
qsw1	QualifiedSwitch (id=30)
output1	FlaggedOption (id=34)
opt2	UnflaggedOption (id=35)
config	JSAPResult (id=37)
help	false
filename	null

Outline

fr.familiar.standalone

 FML

 displayUsage(JSAP, Pr

 S main(String[]) : void

 C FML()

```
129         + e.getLocalizedMessage());
130     }
131   }
132 }
133 FMLShell shell = FMLShell.instantiateStandalone(in);
134
135 boolean verbose = config.getBoolean("verbose");
136 shell.setVerbose(verbose);
137
138 boolean version = config.getBoolean("version");
139 if (version) {
140   System.out.println("version " + FMLShell.FML_VERSION);
141   return;
142 }
143
144 String outputpath = config.getString("output");
145
146 String[] paths = config.getStringArray("paths");
147 for (int i = 0; i < paths.length; ++i) {
148   String path = paths[i];
149   File f = new File(path);
150   if (!f.exists()) {
151     System.err.println("Path " + path + " does not exist");
152     return;
153   }
154 }
```

IntelliJ IDEA File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Reviews.scala - [ScholarServicesScala] - HelloWorldScala - [/Documents/SANDBOX>HelloWorldScala]

Project

HelloWorldScala [ScholarServicesScala] (~/Documents/SANDBOX>HelloWorldScala)

- .idea
- src
 - Reviews
 - ReviewService.scala
 - ScholarServicesScala.iml
 - services.html
 - services.sc
- External Libraries
 - < 1.6 > (/System/Library/Java/JavaVirtualMachin
 - scala-library

services.sc Reviews.scala ReviewService.scala

```
val journals = format(reviews.event.filter(e => e.kind == KJournal))

val confs = format(reviews.event.filter(e => (e.kind == KConference) || (e.kind == KWorkshop)))

val file = new File("services.html")
val doct = DocType("html", PublicID("-//W3C//DTD XHTML 1.0 Strict//EN", "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"), Nil)

val doc =
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">
  <head>
    <title>M. Acher, PhD, Associate Professor (personal webpage)</title>
    <link rel="stylesheet" href="css/myStyle.css" type="text/css" media="screen" />
    <link rel="stylesheet" href="css/nivo-slider.css" type="text/css" />
    <link rel="stylesheet" href="css/jquery.fancybox-1.3.4.css" type="text/css" />
```

Debugger Console Frames

"main"@1 in group "main": RUN...

main():50, Reviews\$

main():-1, Reviews

Variables

- this = {Reviews\$@984}
- args = {java.lang.String[0]@989}
- reviews = {ReviewServices@990}"ReviewServices(List(Venue(SAC'15 (SE),2015,30th ACM Symposium on Applied Computing - Software Engineering (SE) Track,KConference,PCMember,
- event = {scala.collection.immutable.\$colon\$colon@1168}" size = 33
- journals = {scala.xml.Elem@991}"Elem" size = 1
- (0) = {scala.xml.Elem@991}"Elem" size = 1

ETAPE 2 : CHOISIS 3 BONS SOUVENIRS



a97d2829.main.js (index) | a97d2829.main.js:formatted x

```
5058     }, pause: function() {
5059       this.element[this.subwidget]("pause")
5060     }, resume: function() {
5061       this.element[this.subwidget]("resume")
5062     }, getPosition: function() {
5063       return this.element[this.subwidget]("getPosition")
5064     }, getDuration: function() {
5065       return this.element[this.subwidget]("getDuration")
5066     }, playHLS: function(a, c, d, e) {
5067       var f = null;
5068       c && (f = this.element[this.subwidget]("getPosition")), b.debug("play keeping position",
5069     }, preloadHLS: function(a) {
5070       this._hlsArray = a.slice(0), this.element[this.subwidget]("supportDataUri") && (this._pre
5071     }, playHLS: function(a, b, c, d) {
5072       h("wm.send", "event", this._analytics, "navigation", "playVideo"), 1 == d && h("wm.send",
5073       var e = this;
5074       this._preloading ? this._preloading.done(function(a) {
5075
```

Watch Expressions + C

Call Stack □ Async
Not Paused

Scope Variables Not Paused

Breakpoints a97d2829.main.js:formatted:5068
var f = null;

DOM Breakpoints

XHR Breakpoints

Event Listener Breakpoints

Debugging

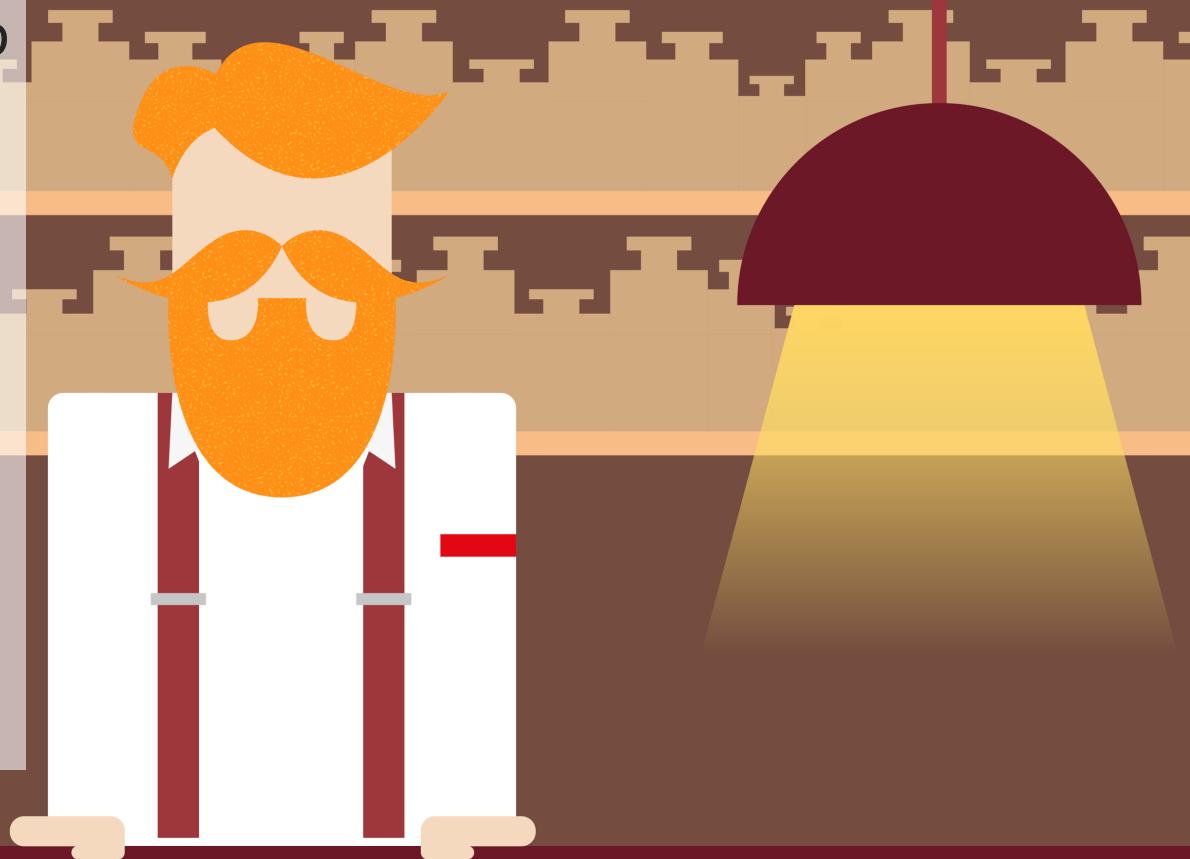
- **Chercher** un chemin/endroit du code qui aboutit à une “failure” (un échec: eg exception)
- Et qui pourrait expliquer un “bug” (une faute)
- Effet vs cause
- Problème: recherche potentiellement gigantesque vu le nombre de lignes de code
- Intelligence du développeur pour restreindre l'espace!

Debugging

- **Chercher:** méthode scientifique
 - Observer (eg message d'erreur)
 - Documenter (eg Github issue)
 - Reproduire (automatiser avec... un test)
 - Analyser, simplifier, émettre des hypothèses
 - Pourquoi cela devrait marcher (Comment comprendre un échec si vous ne savez pas reconnaître un succès?)
 - Analyser les différences
 - Modifier et tester si le problème est résolu
 - Recommencer (en n'oubliant pas ce qui a été appris des précédentes tentatives)

Test

A test engineer walks into
a bar and



— Bill Sempf (@sempf)

A test engineer walks into
a bar and

- orders a beer

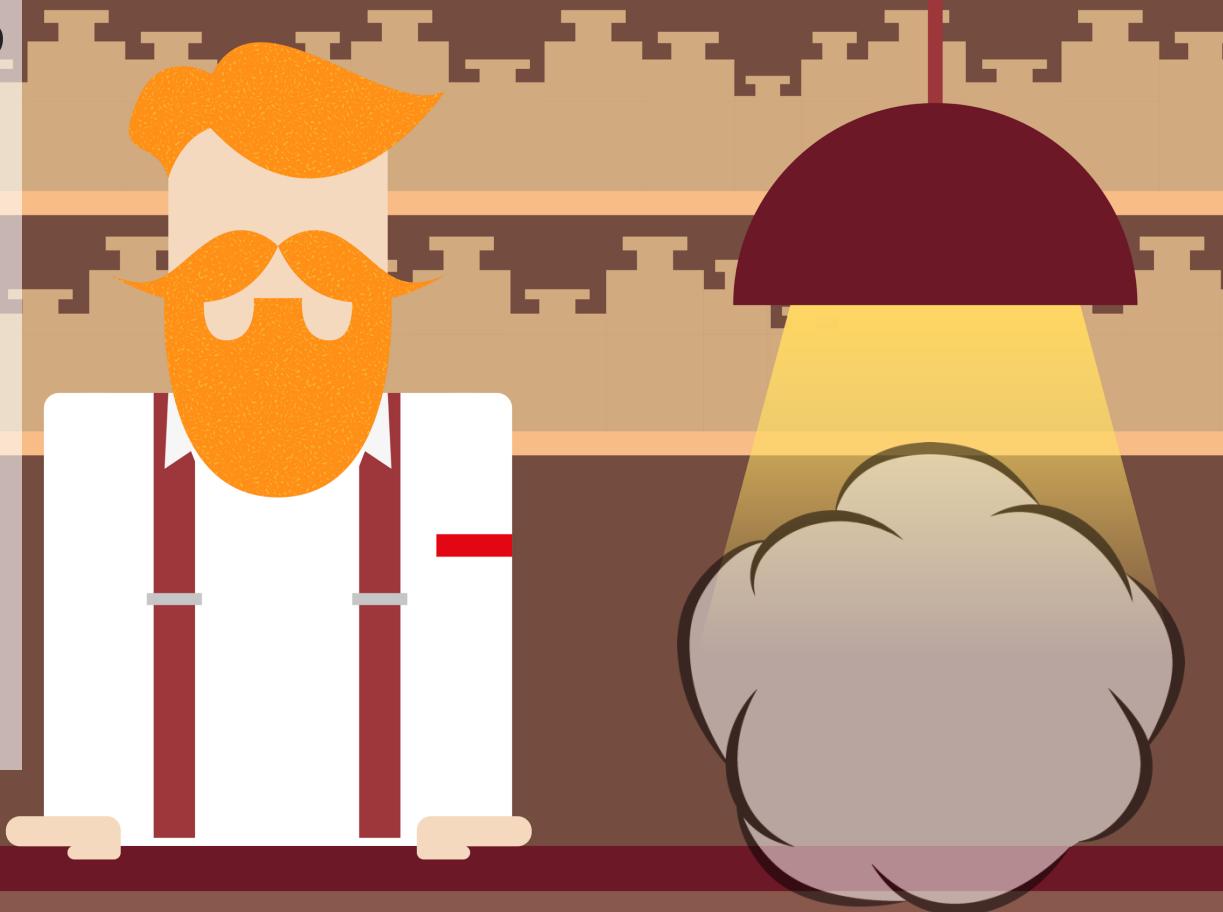
— Bill Sempf (@sempf)



A test engineer walks into
a bar and

- orders a beer
- orders 0 beers

— Bill Sempf (@sempf)

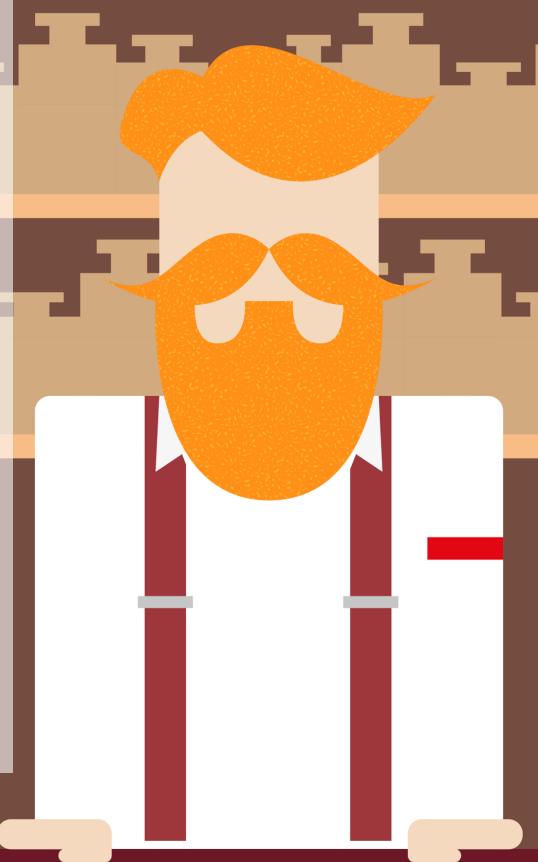


A test engineer walks into
a bar and

- orders a beer
- orders 0 beers
- orders 9999999 beers

— Bill Sempf (@sempf)

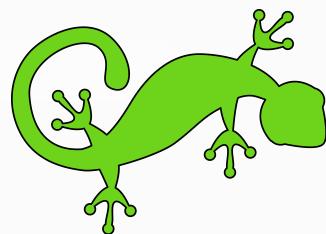
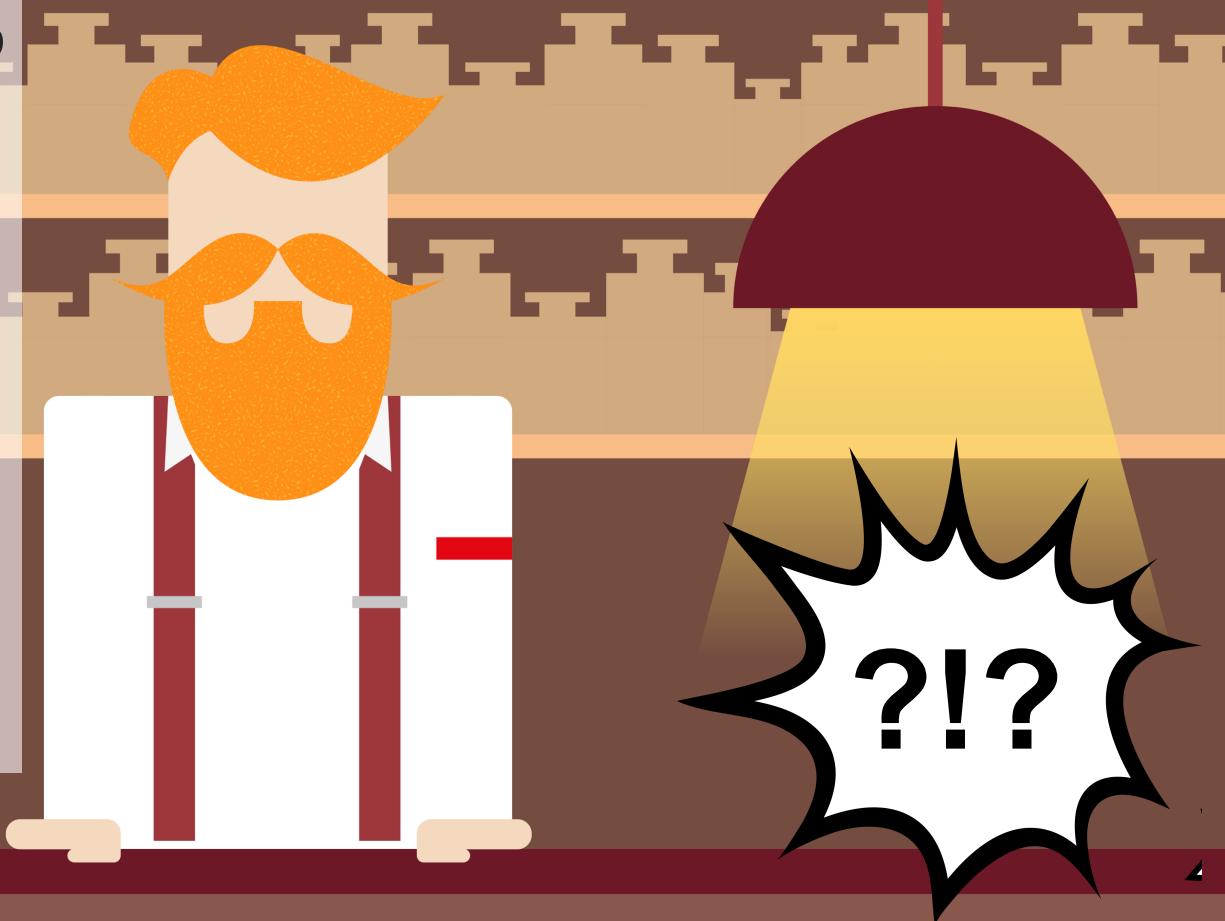
9,999,999



A test engineer walks into
a bar and

- orders a beer
- orders 0 beers
- orders 9999999 beers
- orders a lizard

— Bill Sempf (@sempf)



A test engineer walks into
a bar and

- orders a beer
- orders 0 beers
- orders 9999999 beers
- orders a lizard
- orders -1 beers

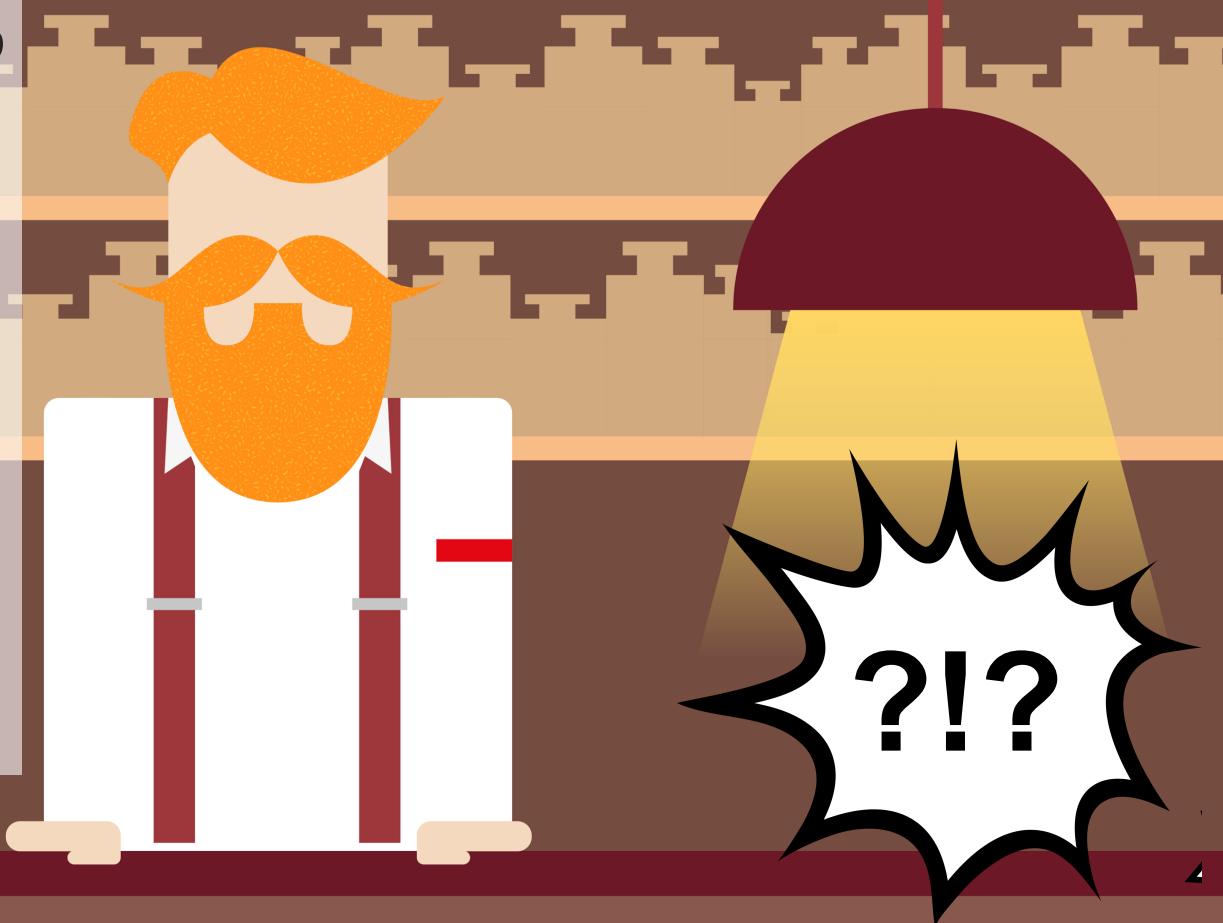
— Bill Sempf (@sempf)



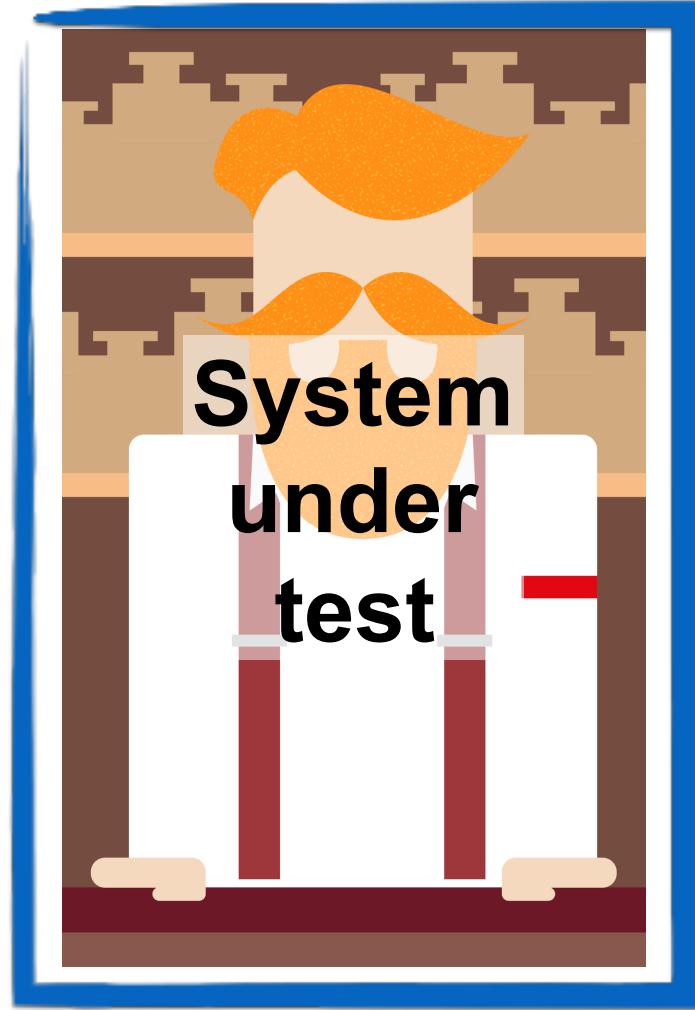
A test engineer walks into
a bar and

- orders a beer
- orders 0 beers
- orders 9999999 beers
- orders a lizard
- orders -1 beers
- orders a "sfdeljknesv"

— Bill Sempf (@sempf)



"sfdeljknesv"



**System
under
test**

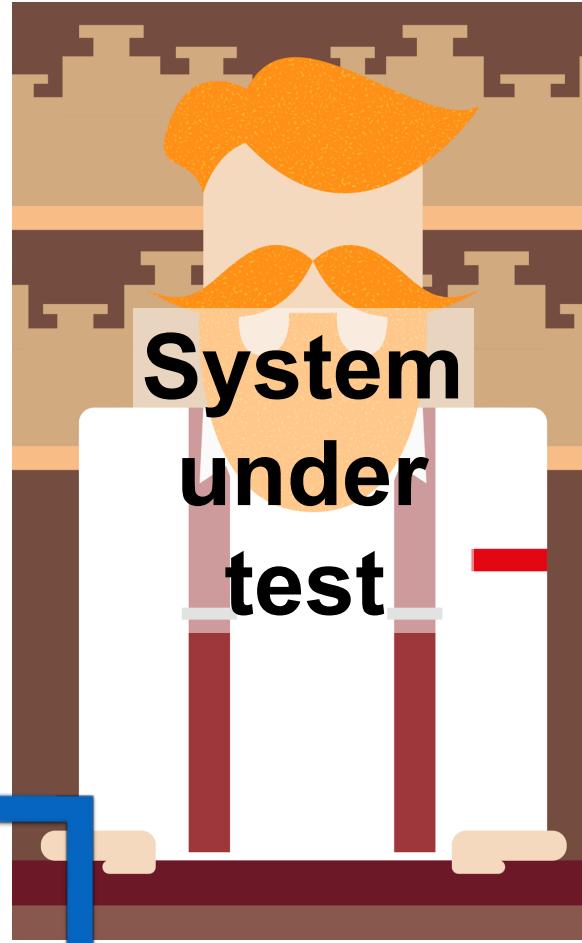


Specification





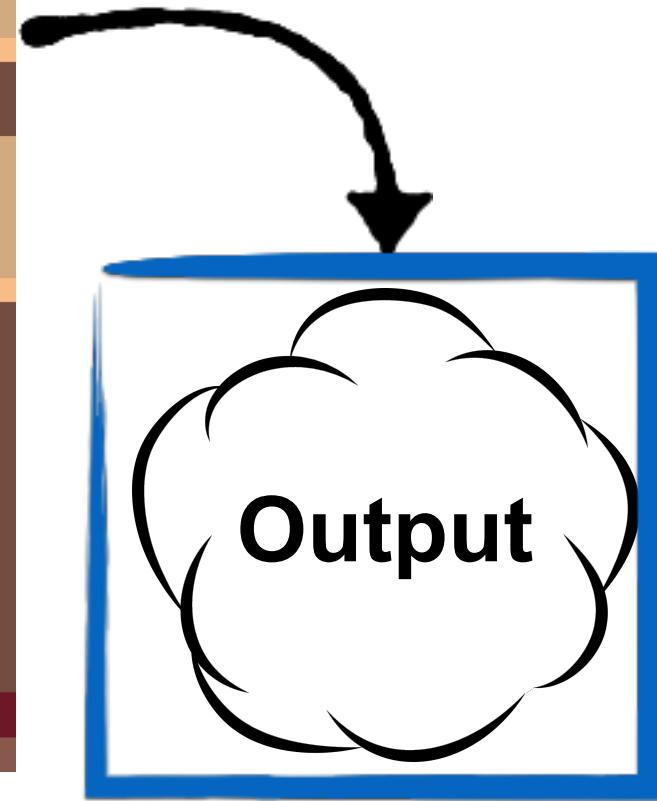
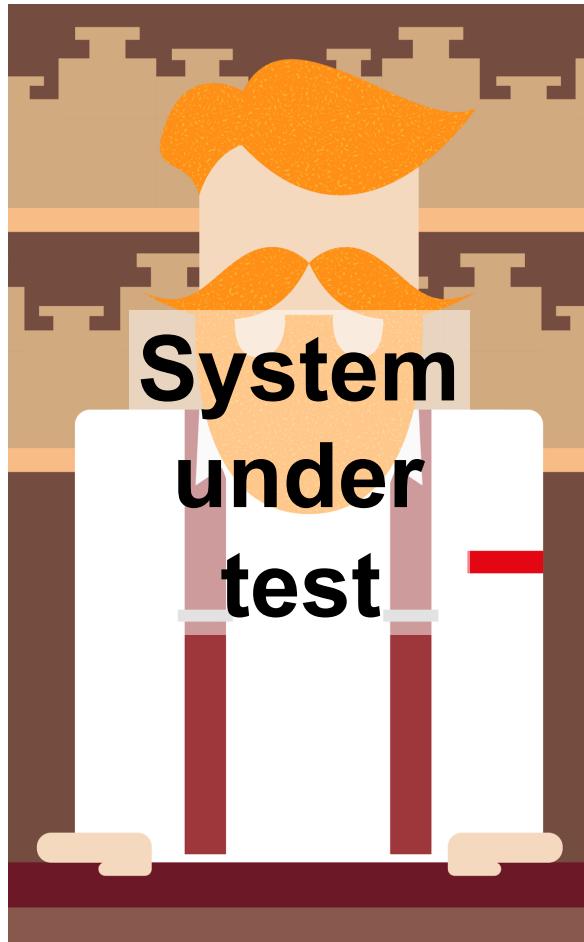
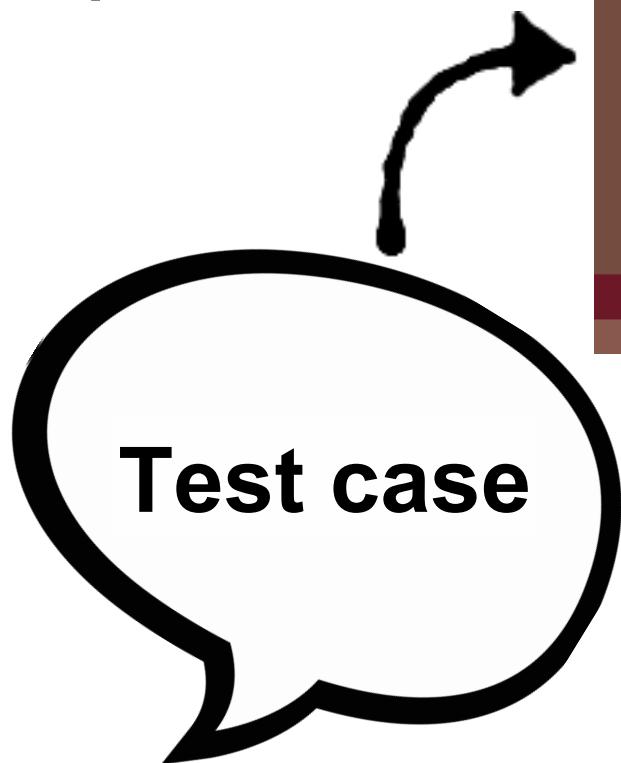
Specification



Test case

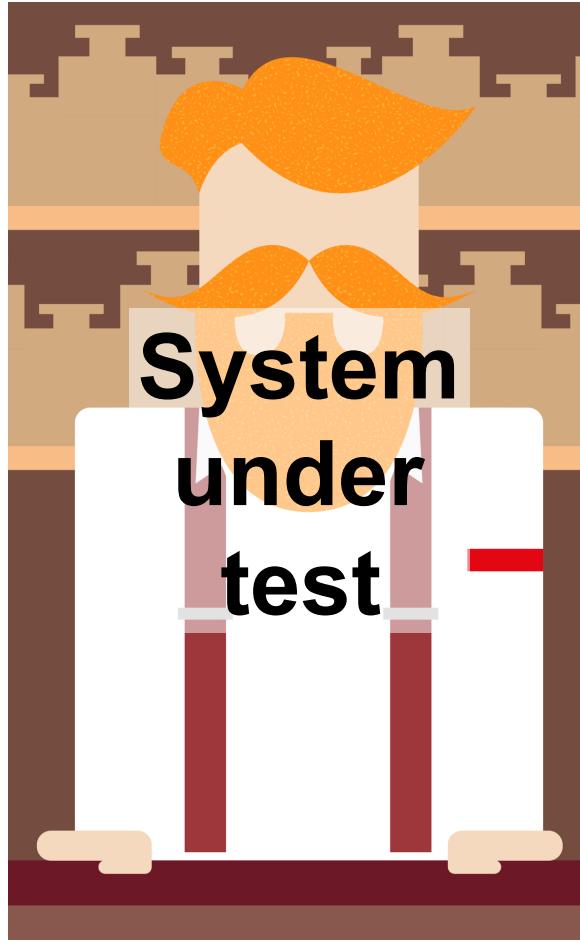


Specification

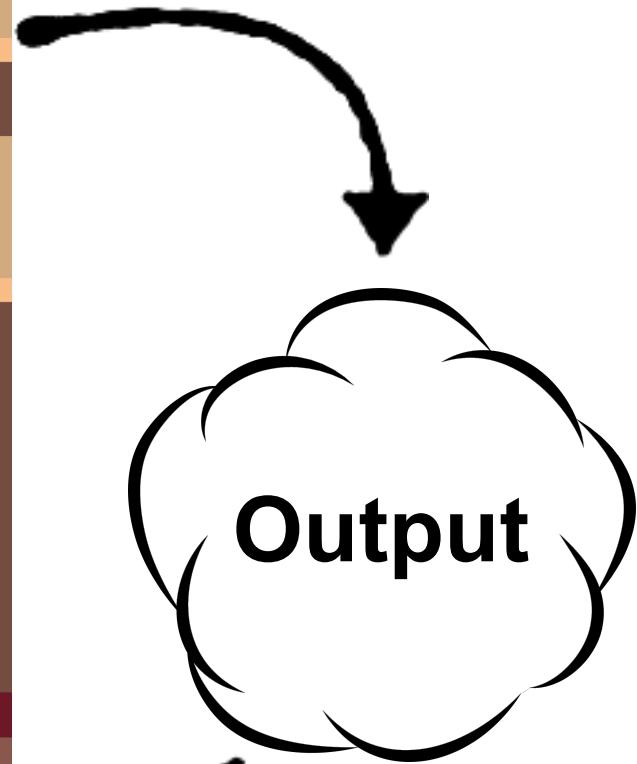




Specification



Test case

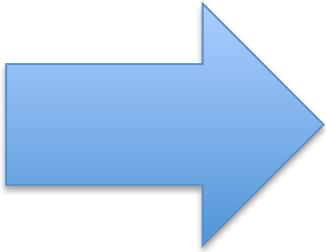


Pass or fail?

Tests automatisés



WIKIPEDIA
The Free Encyclopedia



CSV (Comma Separated Values)

Product	Image process...	Sensor format	Sensor type	Sensor manufac...	Megapixels	Focus points	Metering pixels	Viewfinder cov...
D3X	EXPPEED	Full-frame	CMOS	Sony	24.5	51	1005	100%
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D1X	-	APS-C	CCD	Sony	5.3	5	1005	96%
D1	-	APS-C	CCD	Sony	2.66	5	1005	96%
D4S	EXPPEED 4	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D4	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D3S	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D3	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D2Hs	-	APS-C	JETEL-BCAST	Nikon	4.1	11	1005	100%
D2H	-	APS-C	JETEL-BCAST	Nikon	4.1	11	1005	100%
D1H	-	APS-C	CCD	Sony	2.7	5	1005	96%
D810	EXPPEED 4	Full-frame	CMOS	Sony	36.5	51	91000	100%
D600	EXPPEED 3	Full-frame	CMOS	Sony	36.3	51	91000	100%
D700	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	96%
D700	EXPPEED 4	Full-frame	CMOS	Nikon	24.9	51	91000	100%
Df	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%

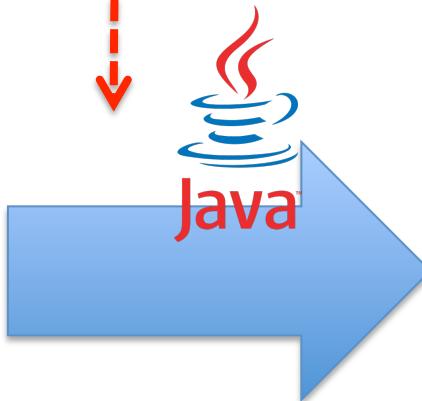
Validation

- Valider ~~manuellement~~, cas par cas
- Valider l'implémentation (tests automatiques)
- Valider les exigences et l'implémentation à chaque itération
 - Sortie de “release” avec procédure de tests automatisée (git + Jenkins + Junit)
 - Validation de chaque release avec le client
- Eliciter et modéliser les exigences/besoins avec le client
 - Qu'est-ce qu'un tableau pertinent?

Tests

(sur les entrées)

(sur la transformation)



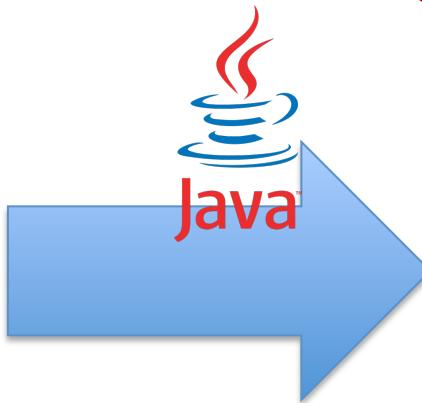
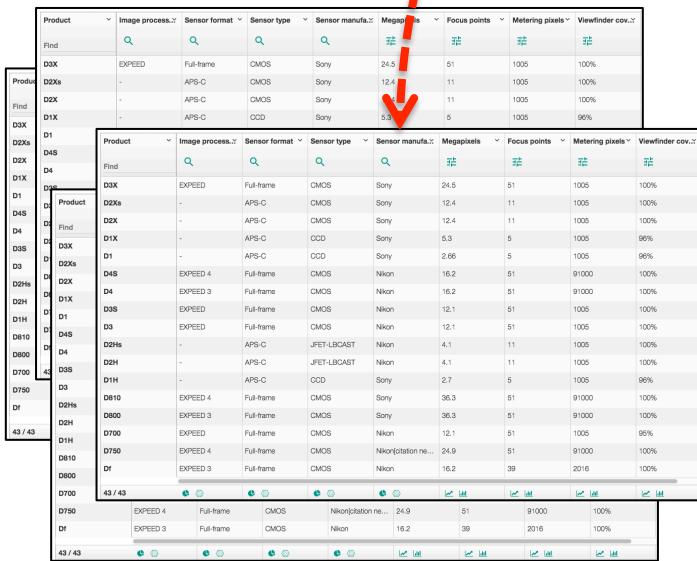
(sur la sortie)

example

CSV

Tests

(sur les entrées)



example

CSV

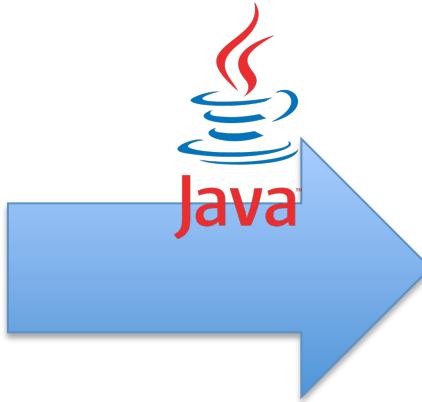
Tests

(sur les entrées)

Product	Image process.:	Sensor format	Sensor type	Sensor manuf.:	Megapixels	Focus points	Metering pixels	Viewfinder cov.:
Find								
D3X	EXPPEED	Full-frame	CMOS	Sony	24.5	51	1005	100%
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D2X	-	APS-C	CMOS	Sony	5.3	5	1005	100%
D1X	-	APS-C	CCD	Sony	2.86	5	1005	96%
D2Xs	EXPPEED 4	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D4S	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D2X	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D1X	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D1	EXPPEED 4	Full-frame	CMOS	Nikon	4.1	11	1005	100%
D2Hs	EXPPEED 3	Full-frame	CMOS	Nikon	2.7	5	1005	96%
D2X	EXPPEED 4	Full-frame	CMOS	Sony	36.3	51	91000	100%
D1H	EXPPEED 3	Full-frame	CMOS	Sony	36.3	51	91000	100%
D800	EXPPEED 4	Full-frame	CMOS	Nikon	24.9	51	91000	100%
D700	EXPPEED 4	Full-frame	CMOS	Nikon	16.2	39	2016	100%
Df	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%
43 / 43								
D7000	EXPPEED 4	Full-frame	CMOS	Nikon (station ne...)	24.9	51	91000	100%
Df	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%
43 / 43								



RuntimeException....

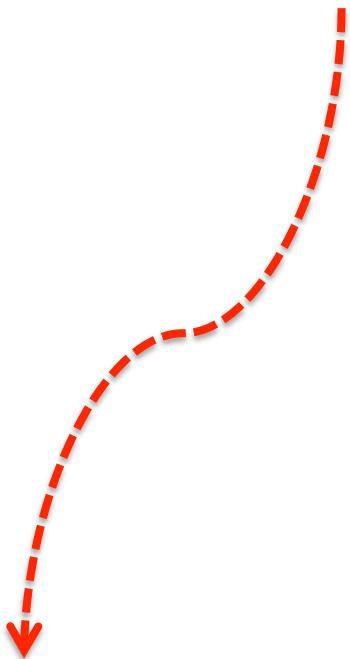


example

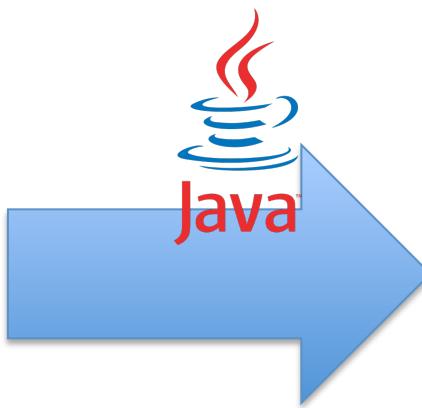
CSV

Tests

(sur les entrées)



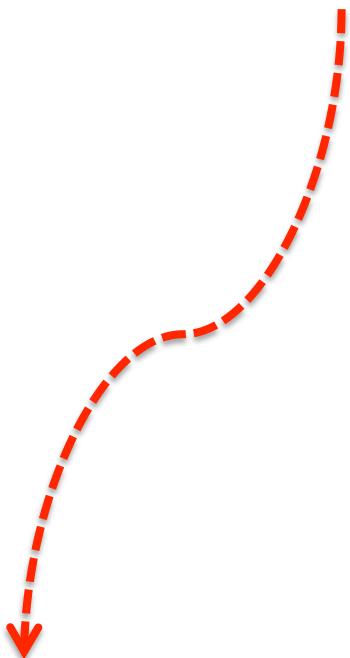
Product	Image process.	Sensor format	Sensor type	Sensor manufac.	Megapixels	Focus points	Metering pixels	Viewfinder cov.
Find								
D3X	EXPPEED	Full-frame	CMOS	Sony	24.5	51	1005	100%
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D1X	-	APS-C	CCD	Sony	5.3	5	1005	98%
D1	-	APS-C	CCD	Sony	2.66	5	1005	96%
D4S	EXPPEED 4	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D4	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D3S	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D3	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D2Hs	-	APS-C	JFET-LBCAST	Nikon	4.1	11	1005	100%
D2H	-	APS-C	JFET-LBCAST	Nikon	4.1	11	1005	100%
D1H	-	APS-C	CCD	Sony	2.7	5	1005	98%
D810	EXPPEED 4	Full-frame	CMOS	Sony	36.3	51	91000	100%
D800	EXPPEED 3	Full-frame	CMOS	Sony	36.3	51	91000	100%
D700	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	98%
D750	EXPPEED 4	Full-frame	CMOS	Nikon (station re...)	24.9	51	91000	100%
Df	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%



F1;f2; , ;
“”.” ”.
‘’ ’ ’

Tests

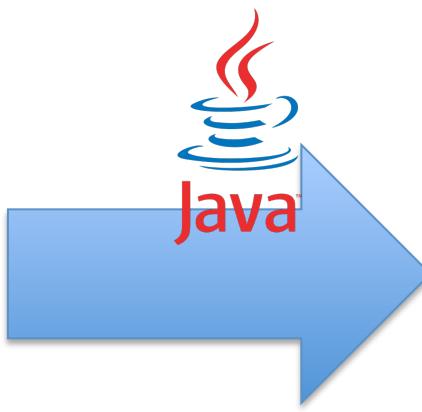
(sur les entrées)



Product	Image process..	Sensor format	Sensor type	Sensor manufac..	Megapixels	Focus points	Metering pixels	Viewfinder cov..
Find								
D3X	EXPPEED	Full-frame	CMOS	Sony	24.5	51	1005	100%
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D1X	-	APS-C	CCD	Sony	5.3	5	1005	98%
D1	-	APS-C	CCD	Sony	2.66	5	1005	96%
D4S	EXPPEED 4	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D4	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D3S	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D3	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D2Hs	-	APS-C	JFET-LBCAST	Nikon	4.1	11	1005	100%
D2H	-	APS-C	JFET-LBCAST	Nikon	4.1	11	1005	100%
D1H	-	APS-C	CCD	Sony	2.7	5	1005	98%
D810	EXPPEED 4	Full-frame	CMOS	Sony	36.3	51	91000	100%
D800	EXPPEED 3	Full-frame	CMOS	Sony	36.3	51	91000	100%
D700	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	98%
D750	EXPPEED 4	Full-frame	CMOS	Nikon (station ..)	24.9	51	91000	100%
Df	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%



f1,f2,f3
v11,v12,v13
v21,v22,v23



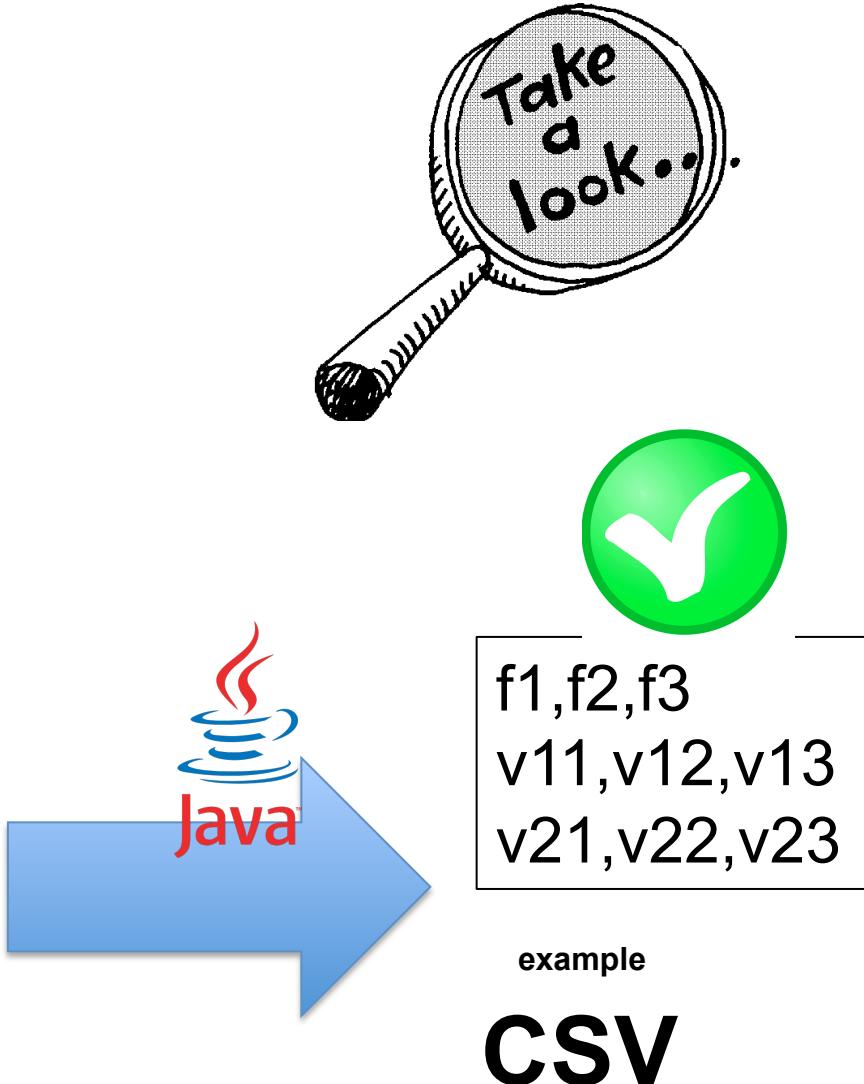
example

CSV

Manual testing is a terrible idea

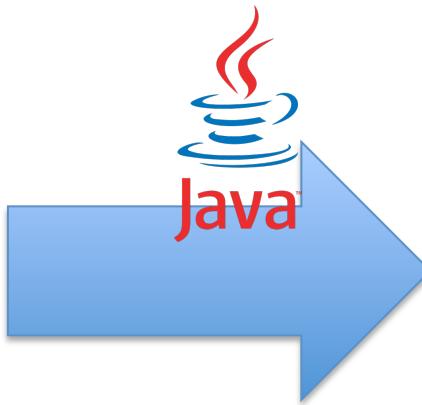
non reproducible; error-prone; time-consuming

Product	Image process.	Sensor format	Sensor type	Sensor manufac.	Megapixels	Focus points	Metering pixels	Viewfinder cov.
Find								
D3X	EXPED	Full-frame	CMOS	Sony	24.5	51	1005	100%
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D1X	-	APS-C	CCD	Sony	5.3	5	1005	98%
D1	-	APS-C	CCD	Sony	2.66	5	1005	96%
D4S	EXPED-4	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D4	EXPED-3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D3S	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D3	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D2Hs	-	APS-C	JFET-LBCAST	Nikon	4.1	11	1005	100%
D2H	-	APS-C	JFET-LBCAST	Nikon	4.1	11	1005	100%
D1H	-	APS-C	CCD	Sony	2.7	5	1005	98%
D810	EXPED-4	Full-frame	CMOS	Sony	36.3	51	91000	100%
D800	EXPED-3	Full-frame	CMOS	Sony	36.3	51	91000	100%
D700	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	98%
D750	EXPED-4	Full-frame	CMOS	Nikon (station re...)	24.9	51	91000	100%
Df	EXPED-3	Full-frame	CMOS	Nikon	16.2	39	2016	100%



You can start with some values/inputs and then (manually) observe

Product	Image process.	Sensor format	Sensor type	Sensor manufa.:	Megapixels	Focus points	Metering pixels	Viewfinder cov.:
Find								
D3X	EXPED	Full-frame	CMOS	Sony	24.5	51	1005	100%
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D1X	-	APS-C	CCD	Sony	5.3	5	1005	98%
D1	-	APS-C	CCD	Sony	2.66	5	1005	96%
D4S	EXPED-4	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D4	EXPED-3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D3S	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D3	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D2Hs	-	APS-C	JFET-LBCAST	Nikon	4.1	11	1005	100%
D2H	-	APS-C	JFET-LBCAST	Nikon	4.1	11	1005	100%
D1H	-	APS-C	CCD	Sony	2.7	5	1005	98%
D810	EXPED-4	Full-frame	CMOS	Sony	36.3	51	91000	100%
D800	EXPED-3	Full-frame	CMOS	Sony	36.3	51	91000	100%
D700	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	98%
D750	EXPED-4	Full-frame	CMOS	Nikon (station re...)	24.9	51	91000	100%
Df	EXPED-3	Full-frame	CMOS	Nikon	16.2	39	2016	100%



f1,f2,f3
v11,v12,v13
v21,v22,v23

example

CSV

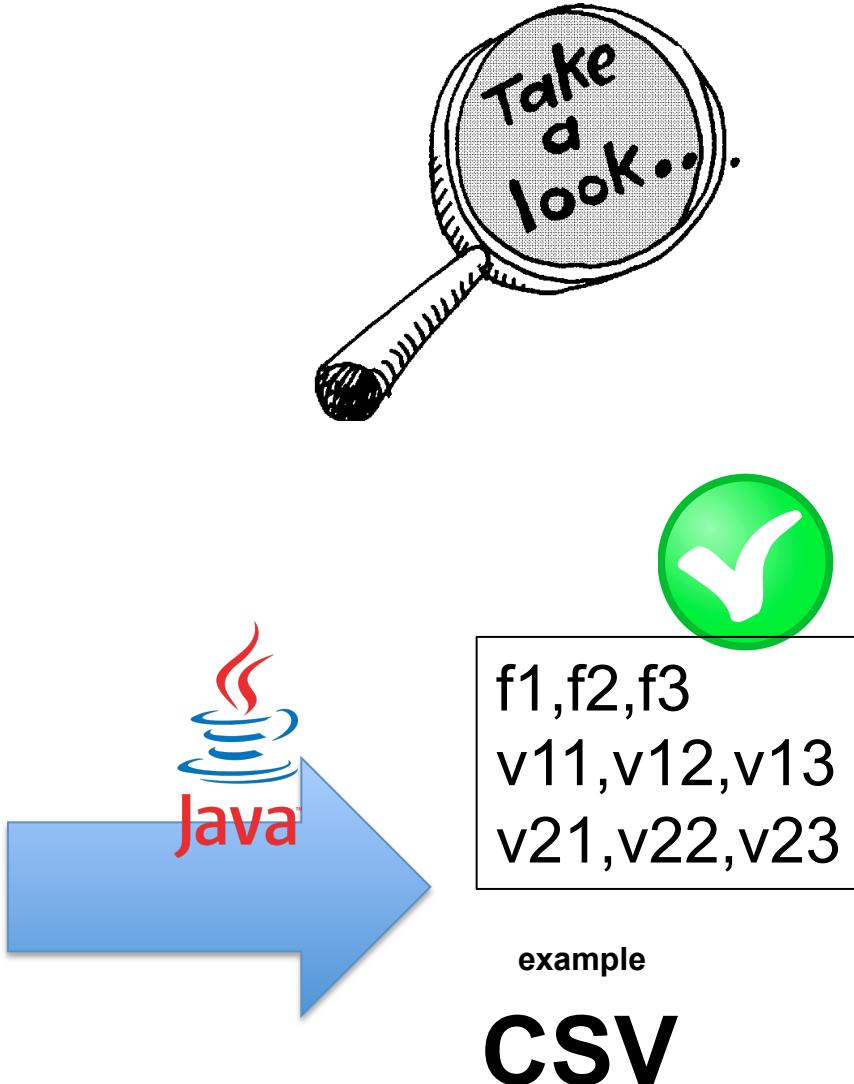


But manual testing is a terrible idea

non reproducible; error-prone; time-consuming

Product	Image process.	Sensor format	Sensor type	Sensor manufac.	Megapixels	Focus points	Metering pixels	Viewfinder cov.
Find								
D3X	EXPPEED	Full-frame	CMOS	Sony	24.5	51	1005	100%
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D1X	-	APS-C	CCD	Sony	5.3	5	1005	98%
D1	-	APS-C	CCD	Sony	2.66	5	1005	96%
D4S	EXPPEED 4	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D4	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D3S	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D3	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D2Hs	-	APS-C	JFET-LBCAST	Nikon	4.1	11	1005	100%
D2H	-	APS-C	JFET-LBCAST	Nikon	4.1	11	1005	100%
D1H	-	APS-C	CCD	Sony	2.7	5	1005	98%
D810	EXPPEED 4	Full-frame	CMOS	Sony	36.3	51	91000	100%
D800	EXPPEED 3	Full-frame	CMOS	Sony	36.3	51	91000	100%
D700	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	98%
D750	EXPPEED 4	Full-frame	CMOS	Nikon (station re...)	24.9	51	91000	100%
Df	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%

43 / 43





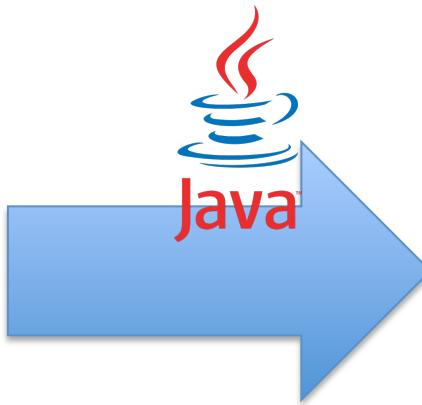
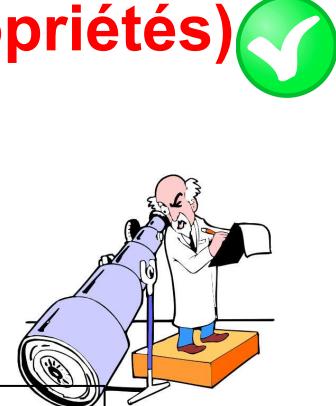
Whenever you are tempted to type something into a print statement or a debugger expression, write it as a test instead.



Tests

(sur les entrées)

Observer par des assertions (vérification de propriétés)



f1,f2,f3
v11,v12,v13
v21,v22,v23

example

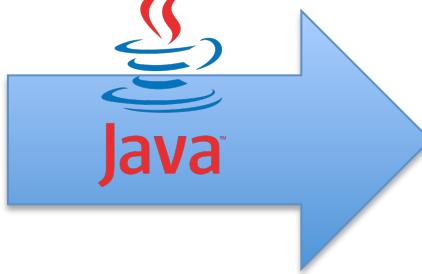
CSV

Tests automatisés

(sur les entrées)

Product	Image process.:	Sensor format	Sensor type	Sensor manuf.:	Megapixels	Focus points	Metering pixels	Viewfinder cov.:	
D3X	EXPED	Full-frame	CMOS	Sony	24.5	51	1005	100%	
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%	
D2X	-	APS-C	CMOS	Sony	5.3	5	1005	100%	
D1X	-	CCD	Sony					98%	
D2Xs	Product	Image process.:	Sensor format	Sensor type	Sensor manuf.:	Megapixels	Focus points	Metering pixels	Viewfinder cov.:
D4S	EXPED	Full-frame	CMOS	Sony	24.5	51	1005	100%	
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%	
D1X	-	APS-C	CCD	Sony	5.3	5	1005	98%	
D1	-	APS-C	CCD	Sony	2.86	5	1005	98%	
D4	EXPED 4	Full-frame	CMOS	Nikon	16.2	51	91000	100%	
D4S	EXPED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%	
D4H	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%	
D2H	-	APS-C	CMOS	Nikon	12.1	51	1005	100%	
D1H	-	APS-C	JET-LBCAST	Nikon	4.1	11	1005	100%	
D1H	-	APS-C	JET-LBCAST	Nikon	4.1	11	1005	100%	
D1H	-	APS-C	CCD	Sony	2.7	5	1005	98%	
D610	EXPED 4	Full-frame	CMOS	Sony	36.3	51	91000	100%	
D610	EXPED 3	Full-frame	CMOS	Sony	36.3	51	91000	100%	
D800	EXPED 3	Full-frame	CMOS	Sony	36.3	51	91000	100%	
D700	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	98%	
D710	EXPED 4	Full-frame	CMOS	Nikon (station ne...)	24.9	51	91000	100%	
D750	EXPED 3	Full-frame	CMOS	Nikon (station ne...)	16.2	39	2016	100%	
Df	EXPED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%	
D7000	EXPED 4	Full-frame	CMOS	Nikon (station ne...)	24.9	51	91000	100%	
Df	EXPED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%	
D7500	EXPED 4	Full-frame	CMOS	Nikon (station ne...)	24.9	51	91000	100%	
Df	EXPED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%	

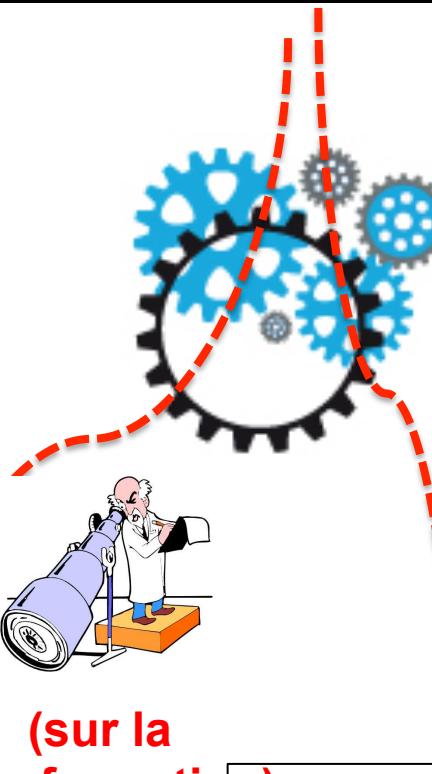
(sur la transformation)



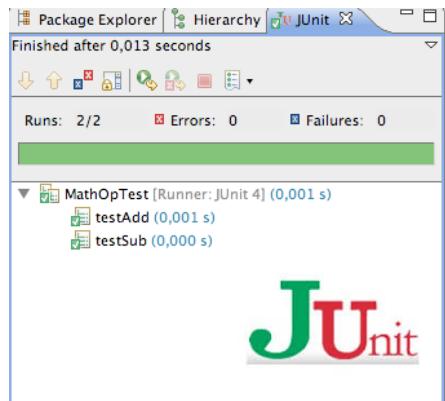
f1,f2,f3
v11,v12,v13
v21,v22,v23

example

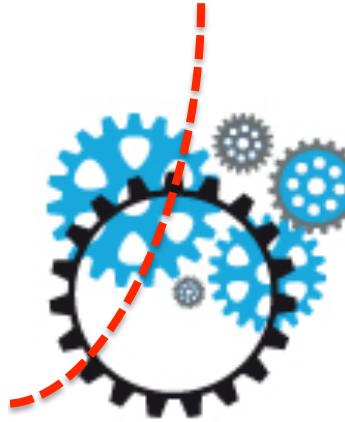
CSV



Tests automatisés

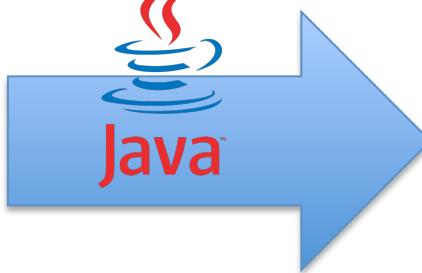


```
// Tests adding a product to the cart.  
public void testProductAdd() {  
    Product book = new Product("Refactoring", 53.95);  
    _bookCart.addItem(book);  
  
    assertTrue(_bookCart.contains(book));  
  
    double expected = 23.95 + book.getPrice();  
    double current = _bookCart.getBalance();  
  
    assertEquals(expected, current, 0.0);  
  
    int expectedCount = 2;  
    int currentCount = _bookCart.getItemCount();  
  
    assertEquals(expectedCount, currentCount);  
}
```



Product	Image process.	Sensor format	Sensor type	Sensor manufa.	Megapixels	Focus points	Metering pixels	Viewfinder cov.
Find								
D3X	EXPPEED	Full-frame	CMOS	Sony	24.5	51	1005	100%
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D1X	-	CCD	Sony		5.3	5	1005	98%
D2Xs	EXPPEED	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D4S	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D2X	EXPPEED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D1X	EXPPEED 3	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D1	EXPPEED 4	Full-frame	CMOS	Nikon	2.86	5	1005	98%
D2Xs	EXPPEED 4	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D4	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D1	EXPPEED 4	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D2H	EXPPEED	Full-frame	CMOS	Nikon	4.1	11	1005	100%
D1H	EXPPEED	Full-frame	CMOS	Nikon	2.7	5	1005	98%
D800	EXPPEED 3	Full-frame	CMOS	Nikon	36.3	51	91000	100%
D700	EXPPEED 3	Full-frame	CMOS	Nikon	36.3	51	91000	100%
D750	EXPPEED 4	Full-frame	CMOS	Nikon(nikonstation ne...)	24.9	51	91000	100%
Df	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%
43 / 43								
D7000	EXPPEED 4	Full-frame	CMOS	Nikon(nikonstation ne...)	24.9	51	91000	100%
Df	EXPPEED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%
43 / 43								

(sur la transformation)



f1,f2,f3
v11,v12,v13
v21,v22,v23

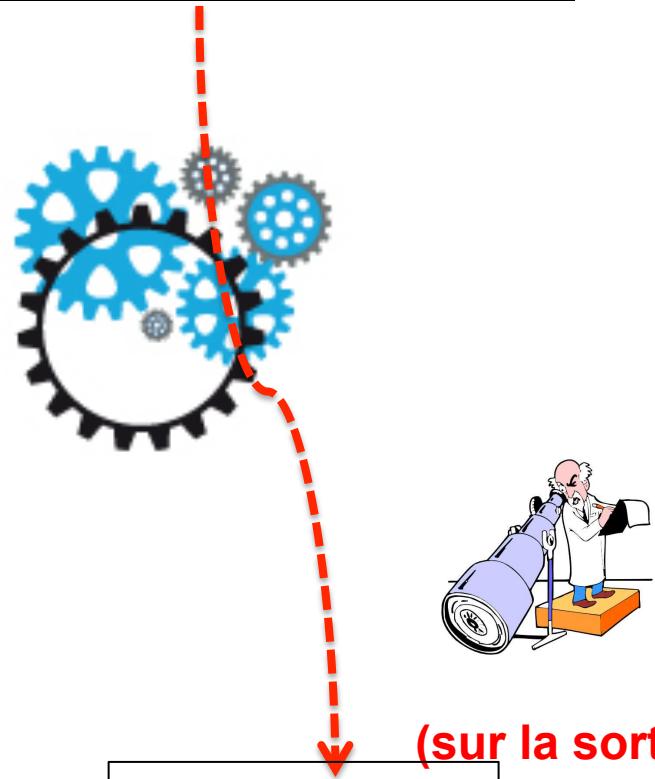
example

CSV

Tests automatisés

(sur les entrées)

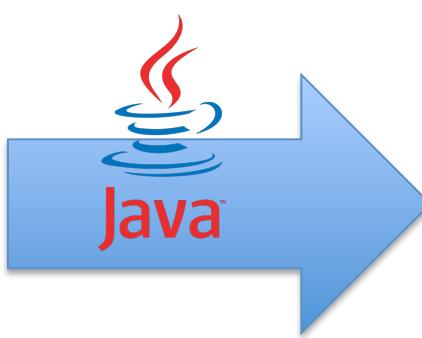
Product	Image process.:	Sensor format	Sensor type	Sensor manuf.:	Megapixels	Focus points	Metering pixels	Viewfinder cov.:
D3X	EXPED	Full-frame	CMOS	Sony	24.5	51	1005	100%
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D1X	-	CCD	Sony		5.3	5	1005	98%
D2Xs	EXPED 4	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D4S	EXPED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D2X	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D1X	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D1	EXPED 4	Full-frame	CMOS	Nikon	2.86	5	1005	98%
D4	EXPED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D1	EXPED	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D1H	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D1H	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D2H	EXPED	Full-frame	CMOS	Nikon	4.1	11	1005	100%
D2Hs	EXPED	Full-frame	CMOS	Nikon	4.1	11	1005	100%
D2H	EXPED 3	Full-frame	CMOS	Nikon	4.1	11	1005	100%
D1H	EXPED 4	Full-frame	CMOS	Nikon	2.7	5	1005	98%
D1H	EXPED 4	Full-frame	CMOS	Nikon	36.3	51	91000	100%
D610	EXPED 4	Full-frame	CMOS	Nikon	36.3	51	91000	100%
D610	EXPED 3	Full-frame	CMOS	Nikon	36.3	51	91000	100%
D800	EXPED 3	Full-frame	CMOS	Nikon	12.1	51	1005	98%
D700	EXPED 4	Full-frame	CMOS	Nikon	24.9	51	91000	100%
D750	EXPED 4	Full-frame	CMOS	Nikon (station ne...)	16.2	39	2016	100%
Df	EXPED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%
D700	EXPED 4	Full-frame	CMOS	Nikon (station ne...)	24.9	51	91000	100%
Df	EXPED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%



f1,f2,f3
v11,v12,v13
v21,v22,v23

example

CSV

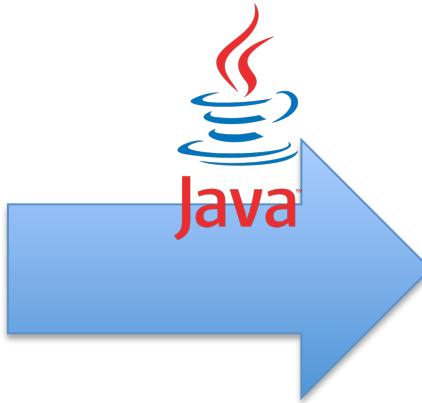


Tests automatisés

(concevoir un ensemble de données en “input” pertinent pour le test et couvrant un maximum de cas)

Product	Image process.:	Sensor format	Sensor type	Sensor manuf.:	Megapixels	Focus points	Metering pixels	Viewfinder cov.:	
Find									
D3X	EXPED	Full-frame	CMOS	Sony	24.5	51	1005	100%	
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%	
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%	
D1X	-	CCD	Sony		5.3	5	1005	98%	
D2Xs	Product	Image process.:	Sensor format	Sensor type	Sensor manuf.:	Megapixels	Focus points	Metering pixels	Viewfinder cov.:
D4S	Find	EXPED	Full-frame	CMOS	Sony	24.5	51	1005	100%
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%	
D1X	-	APS-C	CMOS	Sony	12.4	11	1005	100%	
D1	-	CCD	Sony		5.3	5	1005	98%	
D2Xs	D4	EXPED 4	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D2H	D4S	EXPED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D2Hs	D4	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D1H	D1X	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D1H	D3	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D800	D2Hs	-	APS-C	JET-LBCAST	Nikon	4.1	11	1005	100%
D700	D2H	-	APS-C	JET-LBCAST	Nikon	4.1	11	1005	100%
D750	D3	-	APS-C	Sony	2.7	5	1005	98%	
D750	D800	EXPED 4	Full-frame	CMOS	Sony	36.3	51	91000	100%
D750	D700	EXPED 3	Full-frame	CMOS	Sony	36.3	51	91000	100%
D750	D1H	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	98%
D750	D750	EXPED 4	Full-frame	CMOS	Nikon	24.9	51	91000	100%
D750	Df	EXPED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%
43 / 43									
D750	EXPED 4	Full-frame	CMOS	Nikon	24.9	51	91000	100%	
Df	EXPED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%	
43 / 43									

(vérifier des assertions « génériques » ou bien produire la sortie attendue puis comparaison aka « diff »)



f1,f2,f3
v11,v12,v13
v21,v22,v23

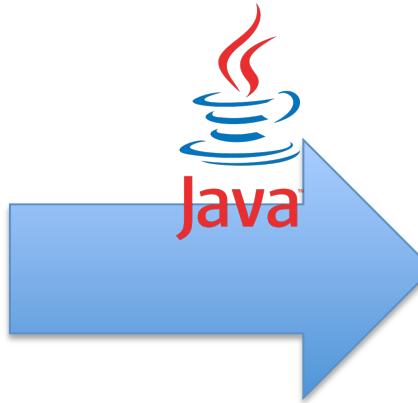
example

CSV

Tests automatisés

NE PAS TESTER VOTRE SOLUTION SUR UNE SEULE PAGE!

(vérifier des assertions « génériques » ou bien produire la sortie attendue puis comparaison aka « diff »)

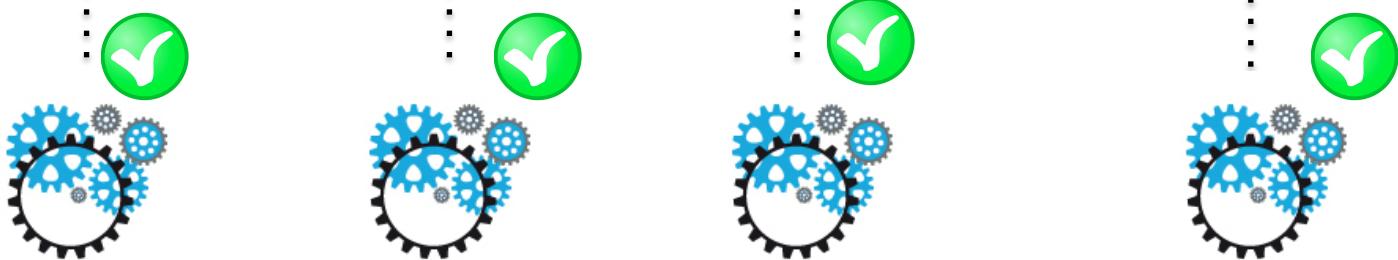


f1,f2,f3
v11,v12,v13
v21,v22,v23

example

CSV

SP (sprints; implémentation)



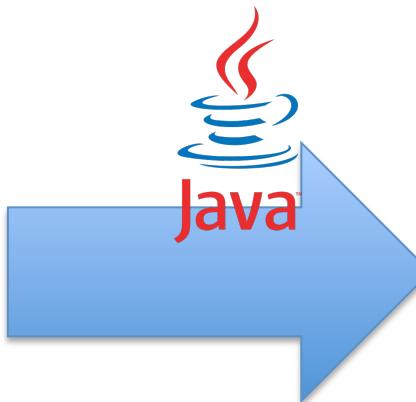
Execute the tests before/after each commit
Don't break (no regression)
Continuous validation

Tests et projets
(bis, discussions)

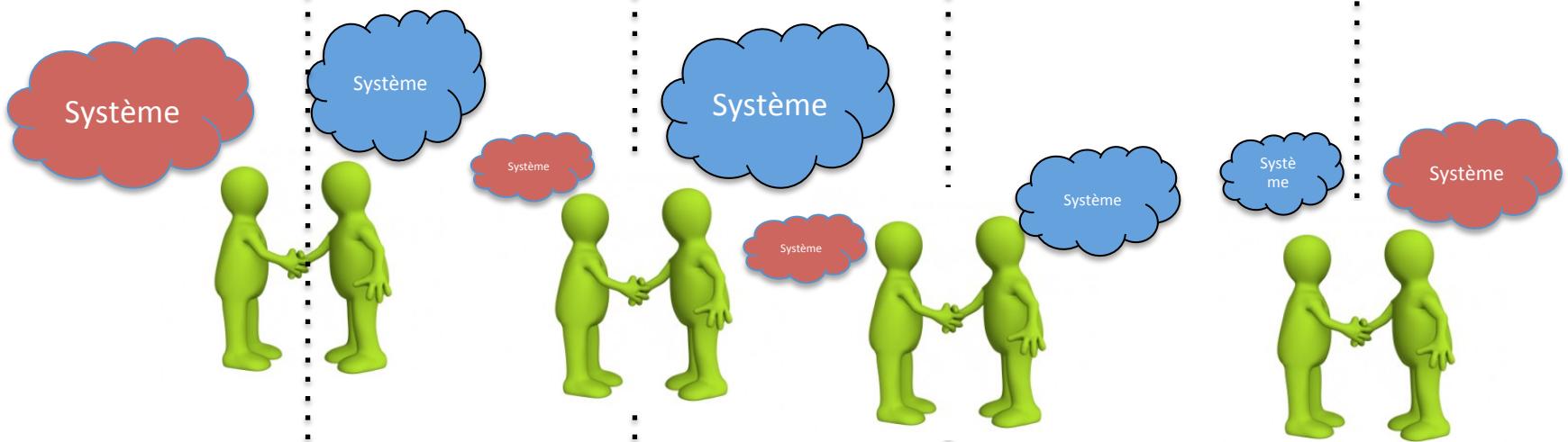
Validation par le client et/ou des utilisateurs

NE PAS TESTER VOTRE SOLUTION SUR UNE SEULE PAGE!

“la stratégie de filtrage n'est pas correcte”
“la librairie XYZ n'est pas adaptée”



EX (exigences; cahier des charges)



Valider à chaque itération avec le client: montrer les modèles, expliquer les choix technologiques, etc.

10 octobre

20 décembre

EX (exigences; cahier
des charges)

SP (sprint;
implémentation)

PR
(présentation)

10 octobre

20 décembre

EX (exigences, cahier
des charges)

SP (sprint;
implémentation)

PR
(présentation)

NON!

Les exigences
évoluent!

10 octobre

20 décembre

EX (exigences, cahier des charges)

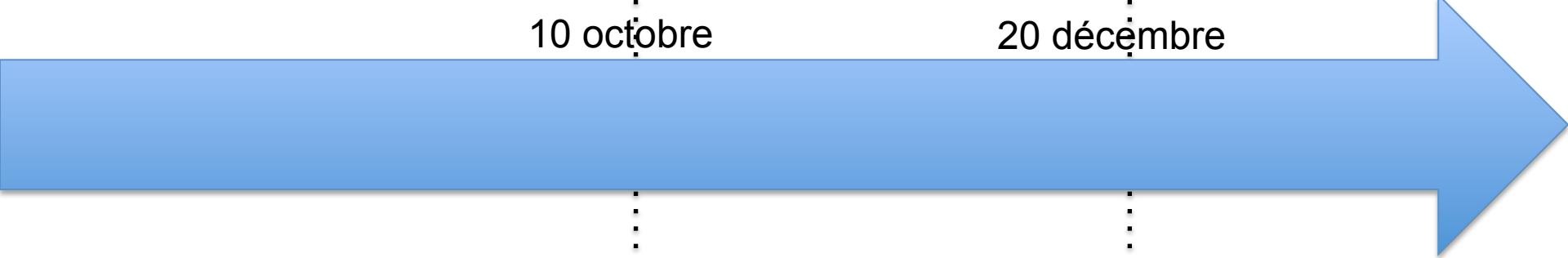
SP (sprint; implémentation)

PR (présentation)

**On ne fixe pas les exigences (même après le 1er délivrable)
Remettre en cause certains éléments du cahier des charges ne serait pas surprenant**

10 octobre

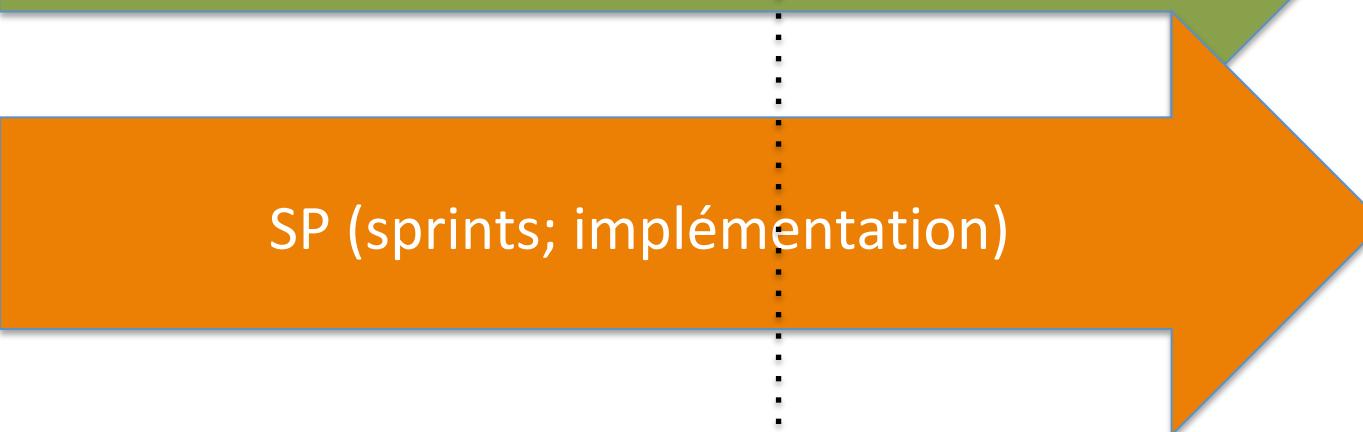
20 décembre



EX (exigences; cahier des charges)



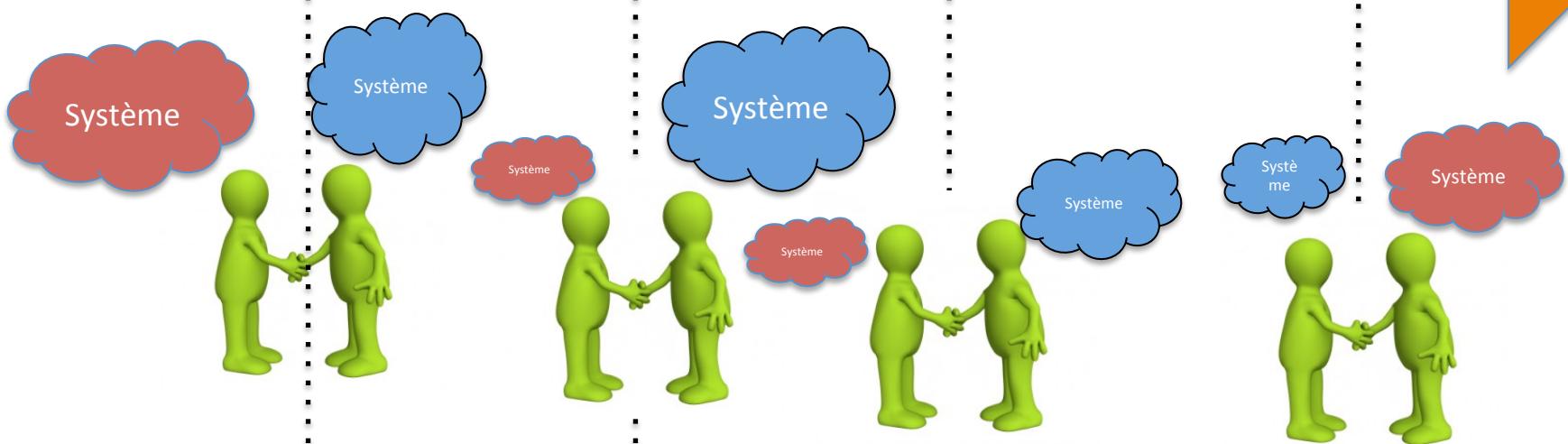
SP (sprints; implémentation)



PR
(présentation)

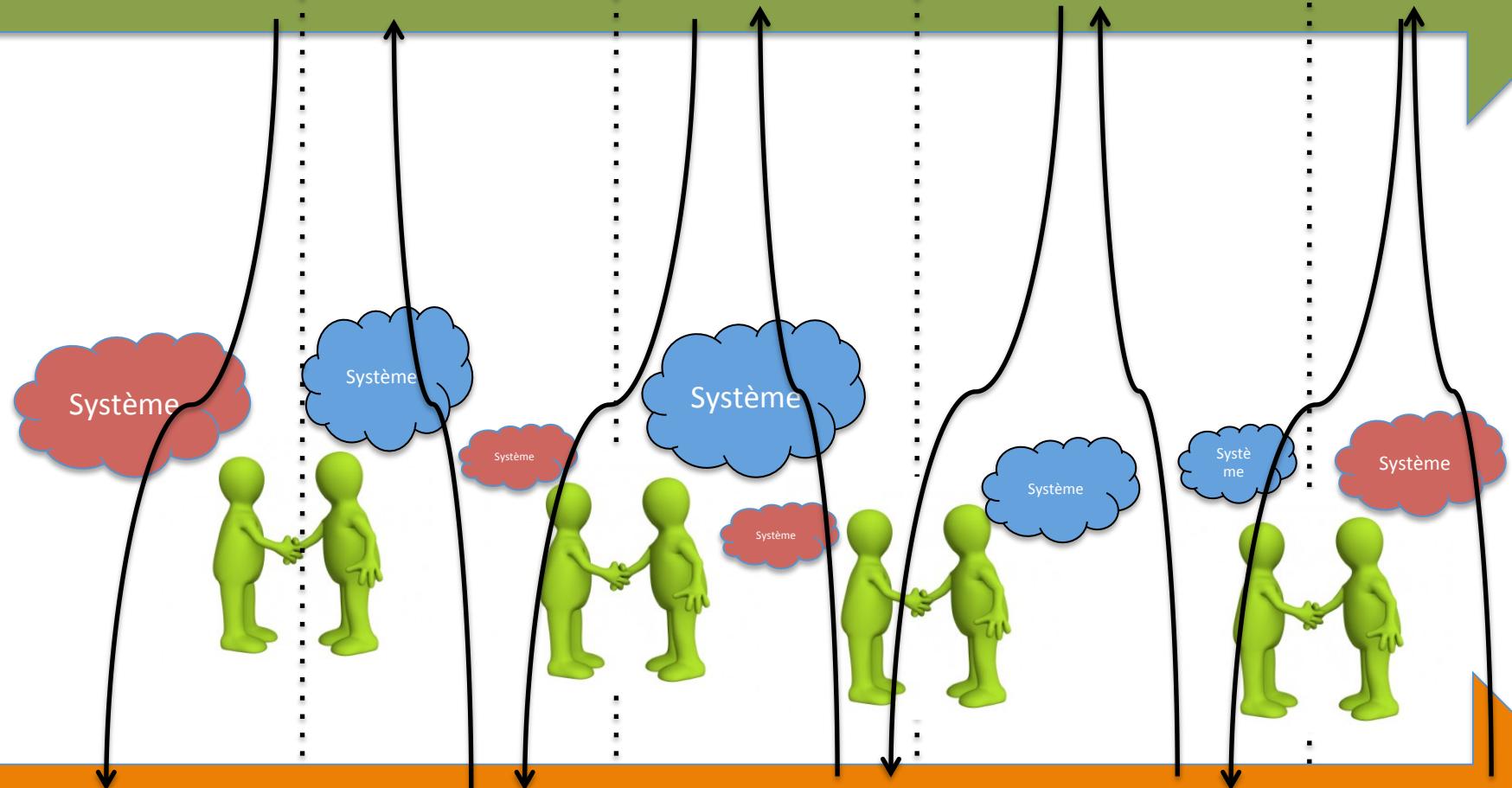
EX (exigences; cahier des charges)

SP (sprints; implémentation)



Valider à chaque itération avec le client: montrer les exigences et l'implémentation (le « produit » en action)

EX (exigences; cahier des charges)



SP (sprints; implémentation)

De `println()`
aux tests automatiques

Travail actuel

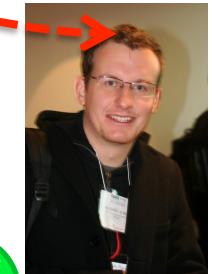
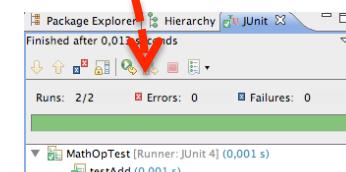
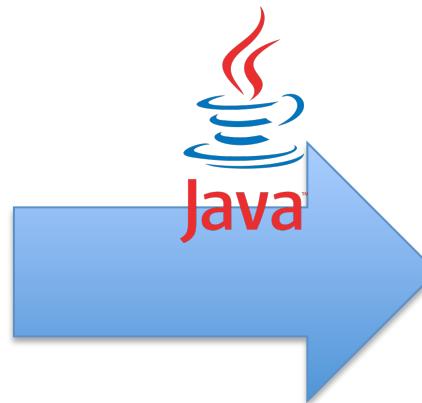
- Installation (ce qu'il est attendu!)
 - **Tests automatiques**
 - Avec Maven
 - Intégration continue
- System.out.println(), logging, debugging
 - Chercher des erreurs (exploration manuelle),
 - Github issues + Test + Fix
- Tests
 - Pour automatiser la validation de votre code (et l'absence de certaines erreurs)

#1 tests automatiques (exhaustif)

#2 validation par le client/des utilisateurs (sampling)

**NE PAS TESTER VOTRE
SOLUTION SUR UNE
SEULE MATRICE!**

Product	Image process.	Sensor format	Sensor type	Sensor manuf.:	Megapixels	Focus points	Metering pixels	Viewfinder cov.:
Find								
D3X	EXPED	Full-frame	CMOS	Sony	24.5	51	1005	100%
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%
D1X	-	APS-C	CCD	Sony	5.3	5	1005	98%
D3X	EXPED 4	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D2Xs	EXPED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%
D4	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D1	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%
D1X	EXPED 3	Full-frame	CMOS	Nikon	4.1	11	1005	100%
D2H	-	APS-C	JET-LBCAST	Nikon	4.1	11	1005	100%
D1H	-	APS-C	JET-LBCAST	Nikon	4.1	11	1005	100%
D800	EXPED 4	Full-frame	CMOS	Sony	2.7	5	1005	98%
D700	EXPED 3	Full-frame	CMOS	Sony	36.3	51	91000	100%
D750	EXPED 4	Full-frame	CMOS	Nikon	24.9	51	91000	100%
Df	EXPED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%
D800	EXPED 4	Full-frame	CMOS	Nikon	24.9	51	91000	100%
D700	EXPED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%
Df	EXPED 4	Full-frame	CMOS	Nikon	16.2	39	2016	100%
D750	EXPED 4	Full-frame	CMOS	Nikon	16.2	39	2016	100%
Df	EXPED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%
43 / 43								
43 / 43								
D750	EXPED 4	Full-frame	CMOS	Nikon	24.9	51	91000	100%
Df	EXPED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%
43 / 43								



f1,f2,f3
v11,v12,v13
v21,v22,v23

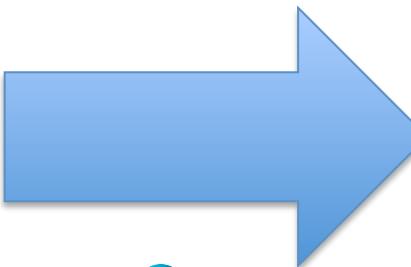
example

CSV

Travail collaboratif et itératif (multi-persons, multi-versions)



Product	Image process.	Sensor format	Sensor type	Sensor manufa.	Megapixels	Focus points	Metering pixels	Viewfinder cov.		
D3X	EXPED	Full-frame	CMOS	Sony	24.5	51	1005	100%		
D2Xs	-	APS-C	CMOS	Sony	12.4	11	1005	100%		
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%		
D1X	-	APS-C	CCD	Sony	5.3	5	1005	96%		
D2Xa	Product	Image process.	Sensor format	Sensor type	Sensor manufa.	Megapixels	Focus points	Metering pixels	Viewfinder cov.	
D1	D3X	EXPED	Full-frame	CMOS	Sony	24.5	51	1005	100%	
D4S	-	APS-C	CMOS	Sony	12.4	11	1005	100%		
D2X	-	APS-C	CMOS	Sony	12.4	11	1005	100%		
D3S	D3X	-	APS-C	CCD	Sony	5.3	5	1005	96%	
D3	D2Xa	-	APS-C	CCD	Sony	2.66	5	1005	96%	
D4	D4S	EXPED 4	Full-frame	CMOS	Nikon	16.2	51	91000	100%	
D2Hs	D4	EXPED 3	Full-frame	CMOS	Nikon	16.2	51	91000	100%	
D2H	D1X	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%	
D1H	D1	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	100%	
D610	D4S	-	APS-C	JET-LBCAST	Nikon	4.1	11	1005	100%	
D800	D3	D2Hs	-	APS-C	JET-LBCAST	Nikon	4.1	11	1005	100%
D750	D2H	-	APS-C	JET-LBCAST	Nikon	4.1	11	1005	100%	
D1H	D1H	-	APS-C	CCD	Sony	2.7	5	1005	96%	
D810	D3	D810	EXPED 4	Full-frame	CMOS	36.3	51	91000	100%	
D2Hs	D800	EXPED 3	Full-frame	CMOS	Sony	36.3	51	91000	100%	
D2H	D700	EXPED	Full-frame	CMOS	Nikon	12.1	51	1005	96%	
D1H	D750	EXPED 4	Full-frame	CMOS	Nikon station ne...	24.9	51	91000	100%	
D810	D1	EXPED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%	
D800	D700	-	EXPED 4	Full-frame	CMOS	36.3	51	91000	100%	
D750	D750	EXPED 4	Full-frame	CMOS	Nikon station ne...	24.9	51	91000	100%	
D1	D800	EXPED 3	Full-frame	CMOS	Nikon	16.2	39	2016	100%	



f1,f2,f3
v11,v12,v13
v21,v22,v23

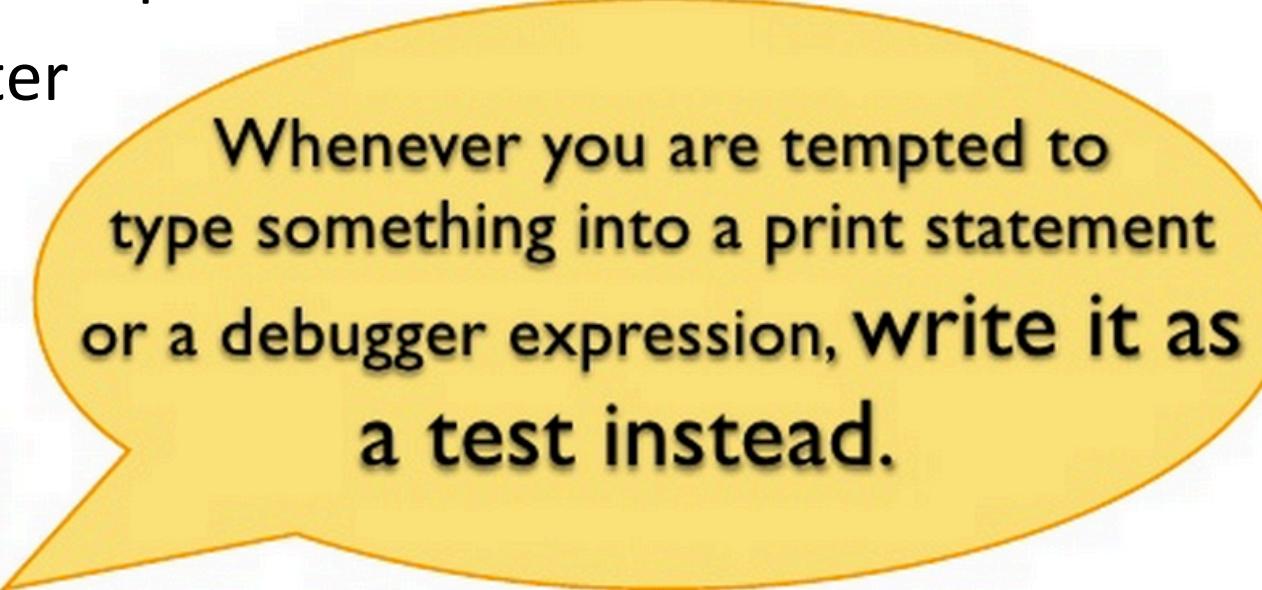
example

CSV



From Logging to Testing

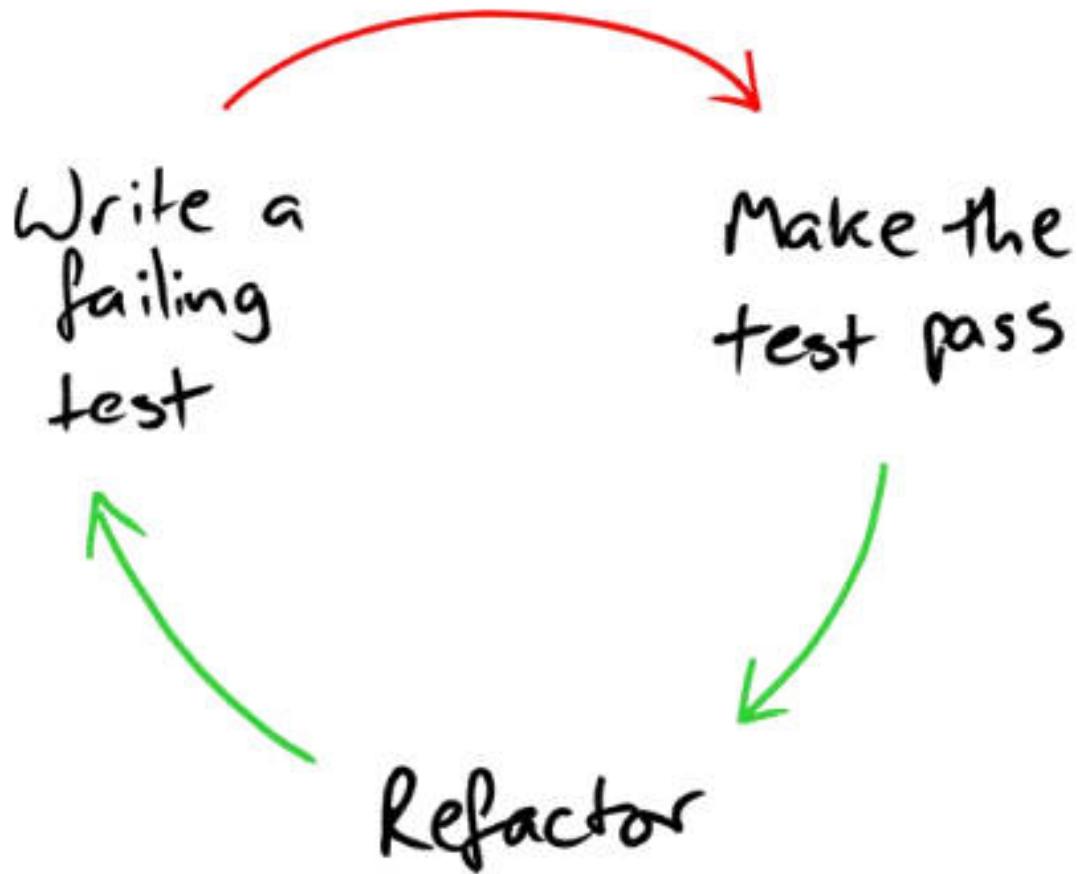
- Testing: “the activity of finding out whether a piece of code produces the intended behavior”
 - Debugging can help
 - Testing is better than debugging



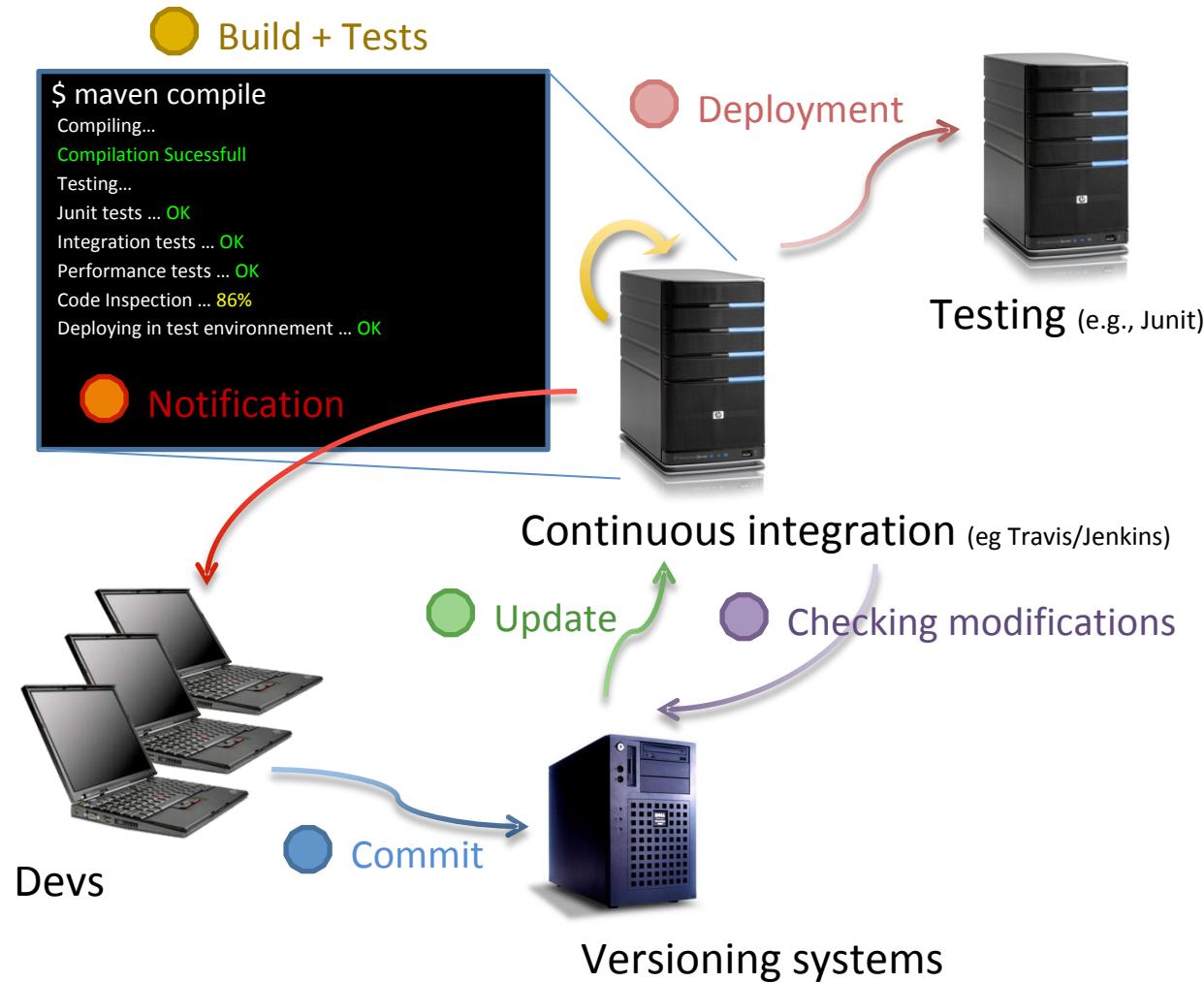
Whenever you are tempted to type something into a print statement or a debugger expression, write it as a test instead.



Automatiser l'exécution
de tests à chaque commit



Usage



Multi-Tools and Languages



Visual Basic



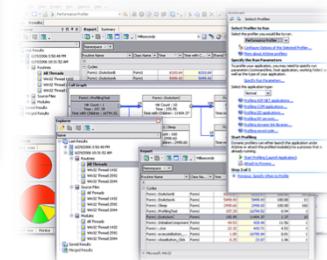
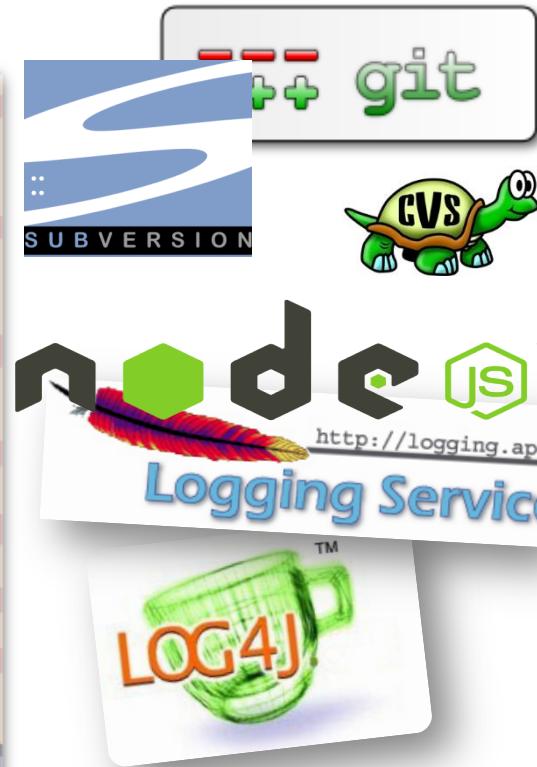
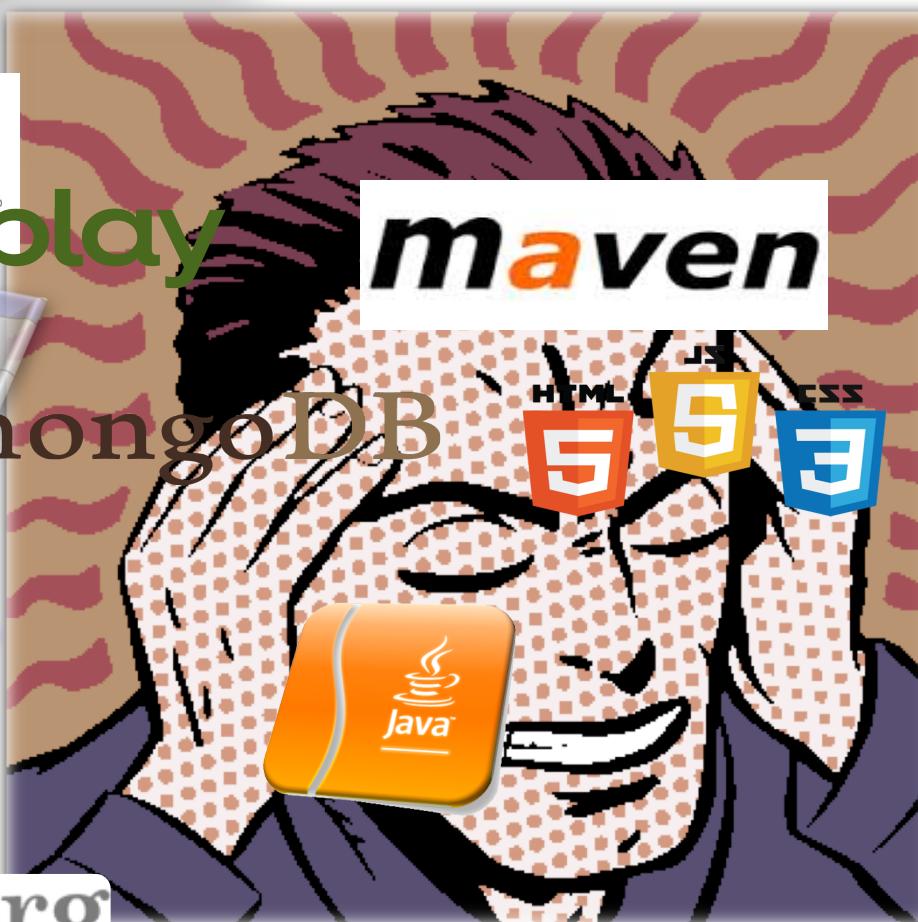
Code::Blocks
Studio



eclipse



JU.org
JUnit



Maven

```
PACKAGE      = package
VERSION       = `date "+%Y.%m%d%"` 
RELEASE_DIR   = ..
RELEASE_FILE  = $(PACKAGE)-$(VERSION)

# Notice that the variable LOGNAME comes from the environment in
# POSIX shells.
#
# target: all - Default target. Does nothing.
all:
    echo "Hello $(LOGNAME), nothing to do by default"
    # sometimes: echo "Hello ${LOGNAME}, nothing to do by default"
    echo "Try 'make help'"

# target: help - Display callable targets.
help:
    egrep "^# target:" [Mm]akefile

# target: list - List source files
list:
    # Won't work. Each command is in separate shell
    cd src
    ls

    # Correct, continuation of the same shell
    cd src; \
    ls
```

Make/Makefile

Original problem: compiling your source code files can be tedious!

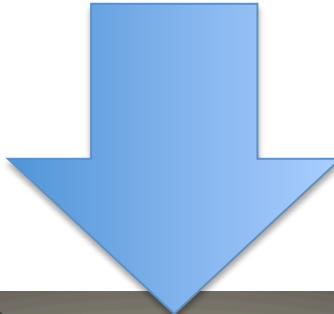
several source files

type the compiling commands **s** everytime

Make for increasing automation, avoiding accidental complexity, and have more flexibility when « compiling » projects

(initial release in 1977)

Make



```
PACKAGE      = package
VERSION      = `date "+%Y.%m%d%" `
RELEASE_DIR  = ..
RELEASE_FILE = $(PACKAGE)-$(VERSION)

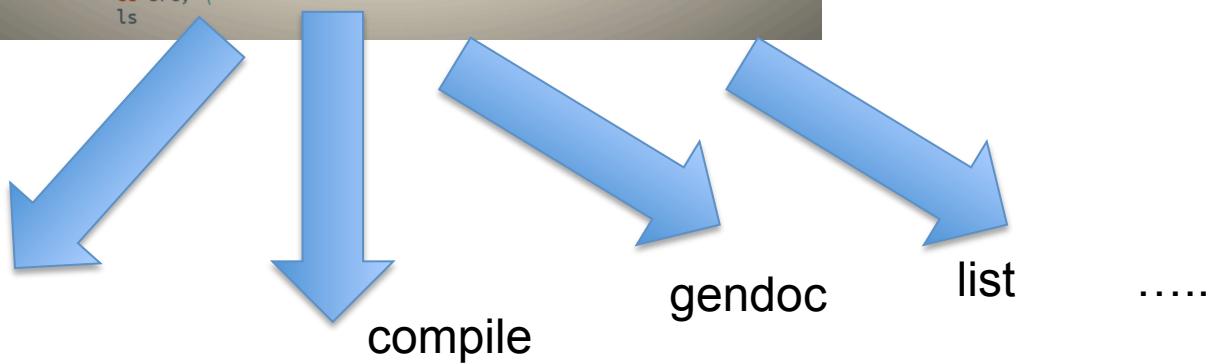
# Notice that the variable LOGNAME comes from the environment in
# POSIX shells.
#
# target: all - Default target. Does nothing.
all:
    echo "Hello $(LOGNAME), nothing to do by default"
    # sometimes: echo "Hello ${LOGNAME}, nothing to do by default"
    echo "Try 'make help'"

# target: help - Display callable targets.
help:
    egrep "^# target:" [Mm]akefile

# target: list - List source files
list:
    # Won't work. Each command is in separate shell
    cd src
    ls

    # Correct, continuation of the same shell
    cd src; \
    ls
```

Makefile



help

compile

gendoc

list

.....

Compile chain

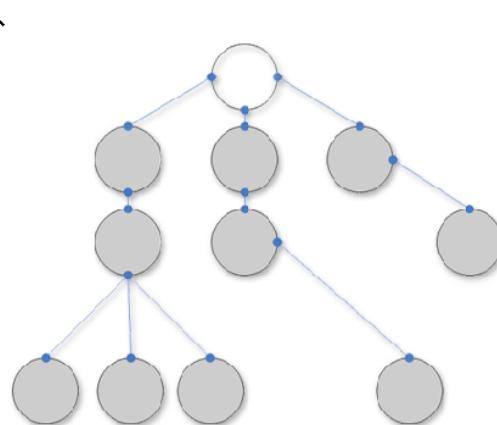
- Sometimes hidden in the IDE
 - But generally speaking, you need to master your “compile” chain
- Tools
 - make, gmake, nmake (Win),
 - Apache ANT, Apache **MAVEN**, Freshmeat 7Bee ...
- To **automate**:
 - pre-compilation, obfuscation, verification
 - generation of .class and .jar
 - normal, tracing, debug, ...
 - documentation generation
 - « stubs » generation (rmic, idl2java, javacard ...)
 - test
 - 3rd party libraries/dependencies
 - ... And a **combination** of all these tasks

What is Maven?

A build tool

```
C:\WINDOWS\system32\cmd.exe
Downloading: http://repo1.maven.org/maven2/org/apache/maven/wagon/1.0-alpha-4/wagon-1.0-alpha-4.pom
3K downloaded
Downloading: http://repo1.maven.org/maven2/org/apache/maven/wagon/wagon-provider-api/1.0-alpha-4/wagon-provider-api-1.0-alpha-4.jar
45K downloaded
Downloading: http://repo1.maven.org/maven2/org/apache/maven/maven-artifact-manager/2.0-alpha-3/maven-artifact-manager-2.0-alpha-3.jar
322K downloaded
[INFO] [install:install]
[INFO] Installing C:\my-app\target\my-app-1.0-SNAPSHOT.jar to C:\Documents and Settings\Administrator.TOSHIBA.m2\repository\com\mycompany\app\my-app\1.0-SNAPSHOT\my-app-1.0-SNAPSHOT.jar
[INFO]
[INFO] BUILD SUCCESSFUL
[INFO]
[INFO] Total time: 47 seconds
[INFO] Finished at: Fri Jun 24 16:24:10 PDT 2005
[INFO] Final Memory: 2M/5M
[INFO]
```

A dependency management tool



A documentation tool

Last Published: Fri Jun 24 22:19:41 EST 2005

Welcome to Maven 2

Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

About Maven 2.0

Get Maven 2.0

Download Maven 2.0 Alpha 3 (1.2Mb)

- System Requirements
- Installation Instructions
- Getting Started

Apply patterns to project build infrastructure

Maven is really a process of applying **patterns** to a build infrastructure in order to provide a coherent view of software projects.

Provides a way to help with managing:

- Builds
- Documentation
- Reporting
- Dependencies
- Software Configuration Management
- Releases

Objectives

- Make the development process visible or transparent
- Provide an easy way to see the health and status of a project
- Decreasing training time for new developers
- Bringing together the tools required in a uniform way
- Preventing inconsistent setups
- Providing a standard development infrastructure across projects
- Focus energy on writing applications

Benefits

- Standardization
- Fast and easy to set up a powerful build process
- Greater momentum vs. Ant - it is now becoming legacy and not moving fast ahead.
- Dependency management (automatic downloads)
- Project website generation, Javadoc
- Repository management
- Extensible architecture

Maven and POM

aka project's configurations

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <version>1.0-SNAPSHOT</version>
  <packaging>jar</packaging>

  <name>Maven Quick Start Archetype</name>
  <url>http://maven.apache.org</url>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.8.2</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

Kind of packaging



Maven facilities and lifecycle

validate: validate the project is correct and all necessary information is available

compile: compile the source code of the project

test: test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed

package: take the compiled code and package it in its distributable format, such as a JAR.

integration-test: process and deploy the package if necessary into an environment where integration tests can be run

verify: run any checks to verify the package is valid and meets quality criteria

install: install the package into the local repository, for use as a dependency in other projects locally

deploy: done in an integration or release environment, copies the final package to the remote repository for sharing with other developers and projects.

clean: cleans up artifacts created by prior builds

site: generates site documentation for this project

Build the Project

```
mvn package
```

Generating the Site

```
mvn site
```

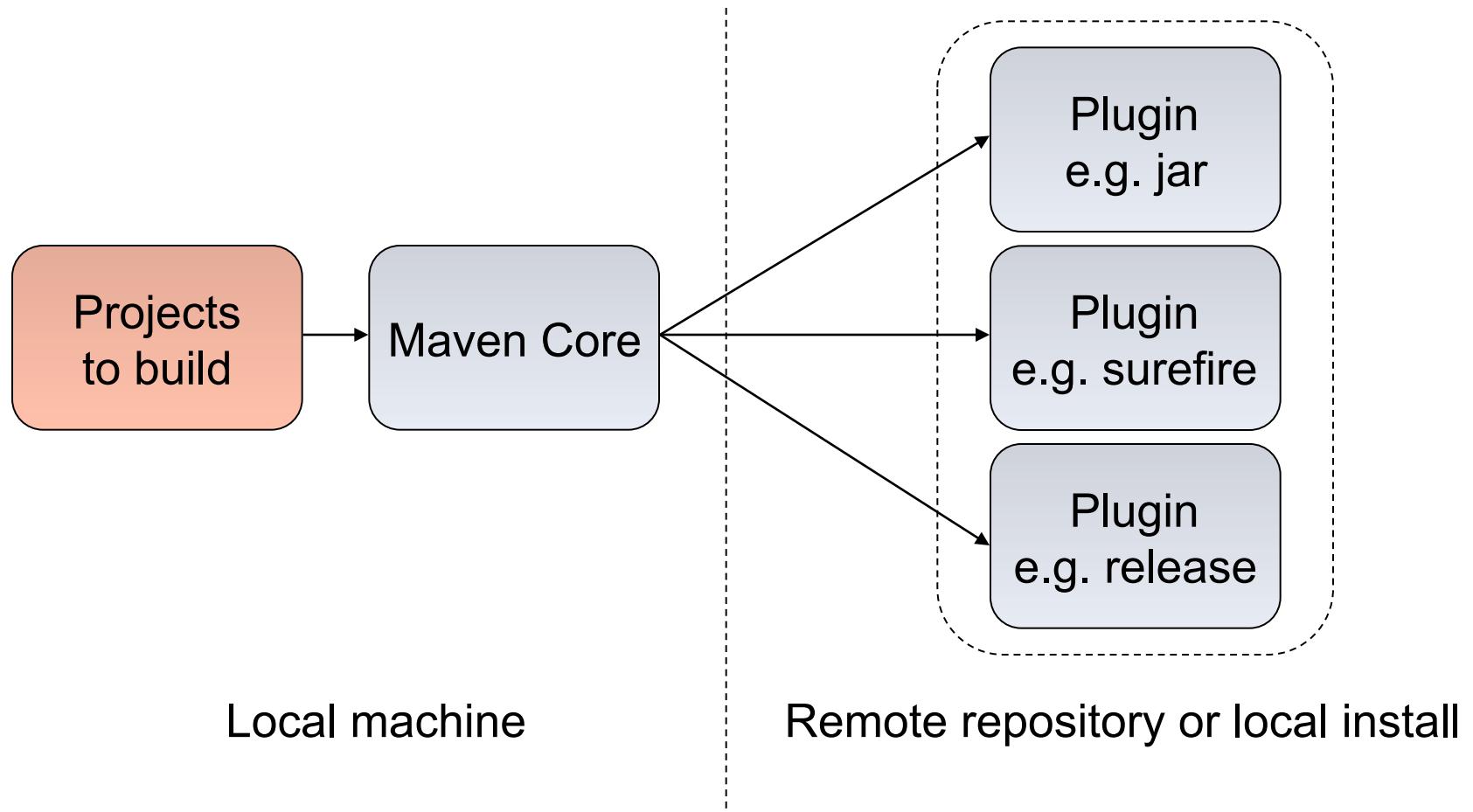
Maven

- Abstract project model (POM)
 - Object oriented, inheritance
 - Separation of concerns
- Default lifecycle
 - Default state (goals) sequence
 - plugins depend on states
- Give a project « standard » structure
 - Standard naming conventions
 - Standard lifecycle
- Automatic handling of dependencies between projects
 - Including updates
- Project repositories
 - public or private, local or remotes
 - caching and proxy
- Extensible via external plugins

Maven plugins

- Core
 - clean, compiler, deploy, install, resources, site, surefire, verifier
- Packaging
 - ear, ejb, jar, rar, war, bundle (OSGi)
- Reporting
 - changelog, changes, checkstyle, clover, doap, docck, javadoc, jxr, pmd, project-info-reports, surefire-report
- Tools
 - ant, antrun, archetype, assembly, dependency, enforcer, gpg, help, invoker, one (interop Maven 1), patch, plugin, release, remote-resource, repository, scm
- IDEs
 - eclipse, netbeans, idea
- Others
 - exec, jdepend, castor, cargo, jetty, native, sql, taglist, javacc, obr
 - ...

Maven Architecture



Common project metadata format

- POM = Project Object Model = pom.xml
- Contains metadata about the project
 - Location of directories, Developers/Contributors, Issue tracking system, Dependencies, Repositories to use, etc
- Example:

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>org.codehaus.cargo</groupId>
  <artifactId>cargo-core-api-container</artifactId>
  <name>Cargo Core Container API</name>
  <version>0.7-SNAPSHOT</version>
  <packaging>jar</packaging>
  <dependencies/>
  <build/>
  [...]
```

Minimal POM

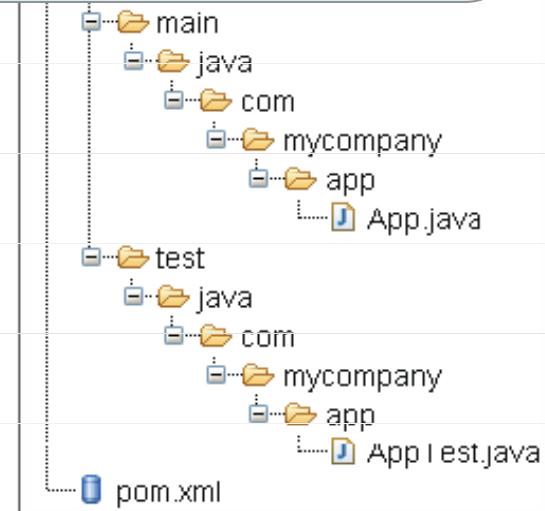
Use Inheritance

Standard directory organization

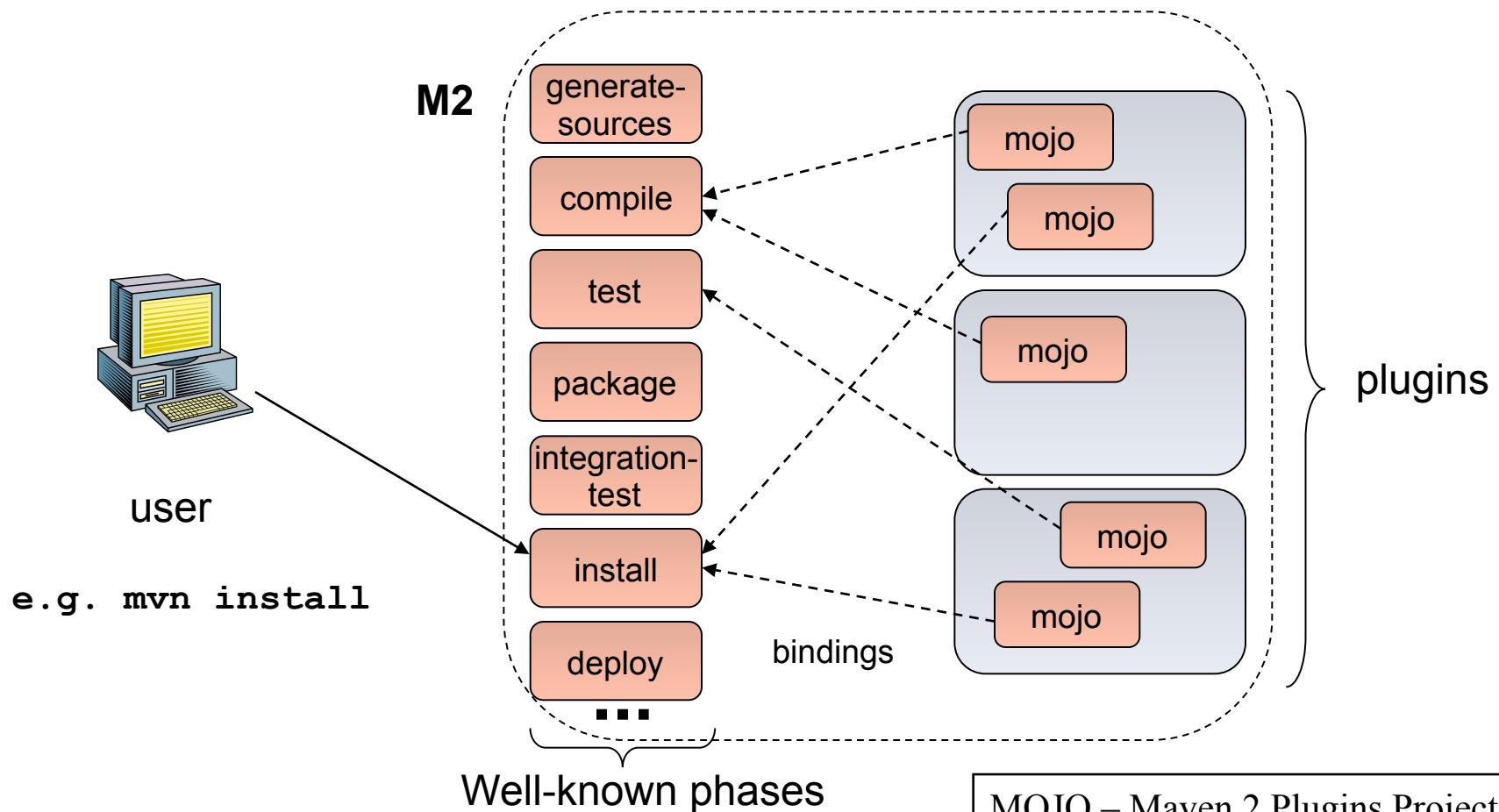
- Having a common directory layout would allow for users familiar with one Maven project to quickly understand another Maven project

src/main/java	
src/main/resources	
src/main/filters	
src/main/assembly	Assembly descriptor files
src/main/config	Configuration files
src/main/webapp	Web application resources
src/test/java	Test source code
src/test/resources	Test resource files
src/test/filters	Test resource filter files
src/site	Site
LICENSE.txt	Project's license
README.txt	Project's readme

Convention over configuration

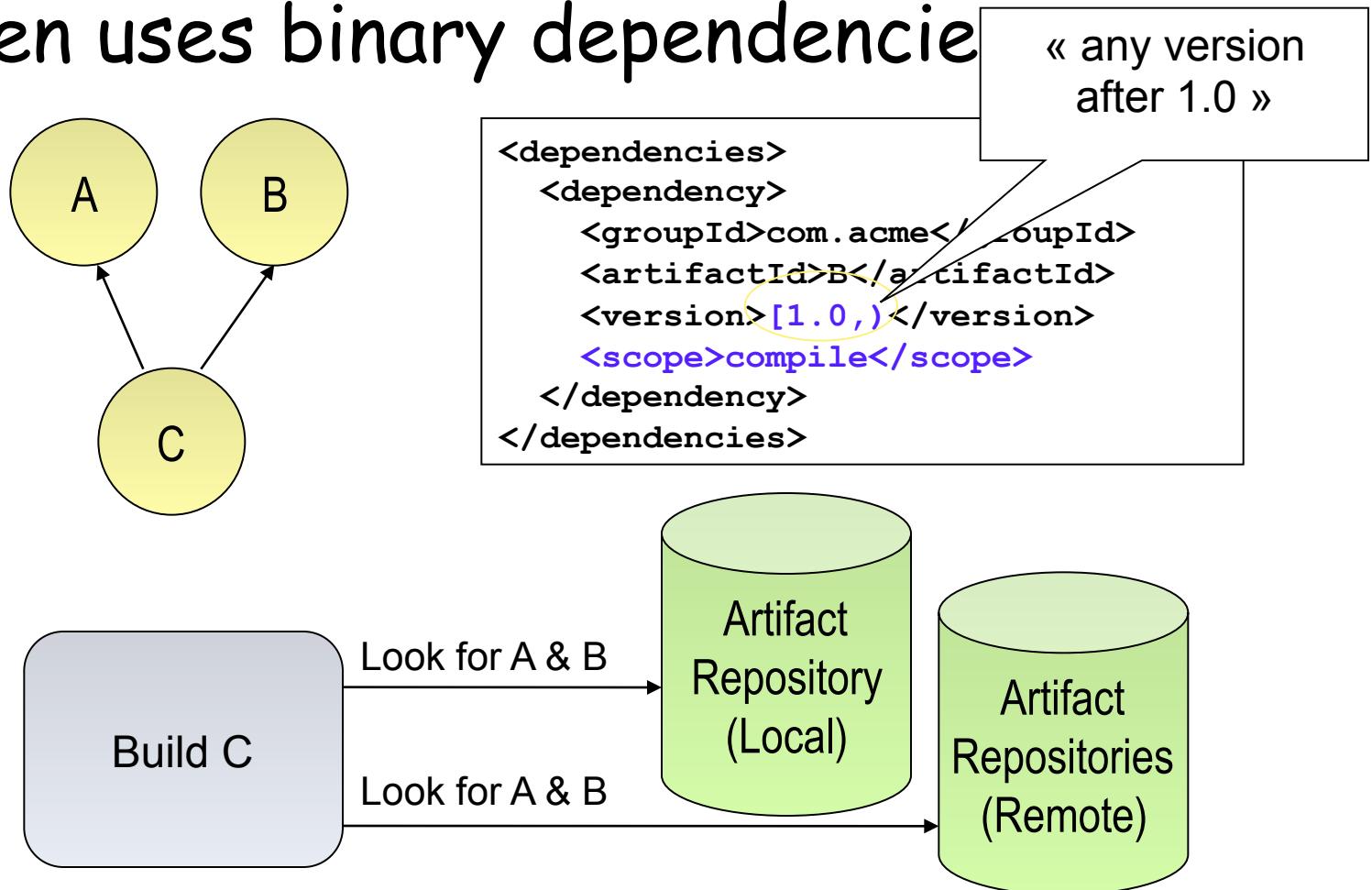


Common way to build applications



Dependency management (1/2)

- Maven uses binary dependencies



Maven plugin for JAVA IDE

- Maven plugins exists for
 - Eclipse
 - IntelliJ
 - NetBeans
 - ...

Overview of common Goals

- **clean** - clean the current project
- **validate** - validate the project is correct and all necessary information is available
- **compile** - compile the source code of the project
- **test** - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed
- **package** - take the compiled code and package it in its distributable format, such as a JAR
- **integration-test** - process and deploy the package if necessary into an environment where integration tests can be run
- **install** - install the package into the local repository, for use as a dependency in other projects locally
- **deploy** - done in an integration or release environment, copies the final package to the remote repository for sharing with other developers and projects

More stuff

- Automatically generate reports, diagrams, and so on through Maven / the project site
- Internationalization - create different language project websites
- Create projects within projects (more pom.xml files inside sub dirs\jars), with different build stats and so on
- Maven can make .war files, EJBs, etc.

Using Maven Plugins

- Whenever you want to customise the build for a Maven project, this is done by adding or reconfiguring plugins
- For example, configure the Java compiler to allow JDK 5.0 sources
- Plugins in Maven 3.0 look much like a dependency

```
...
<build>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <configuration>
        <source>1.5</source>
        <target>1.5</target>
      </configuration>
    </plugin>
  </plugins>
</build>
...
```

Maven Plugins

- **AlmostPlainText**
- **Maven Axis**
- **Maven Cobertura**
- **Maven DB2**
- **Dbunit**
- **Debian Package**
- **Maven DotUml**
- **Doxxygen**
- **Maven Files**
- **FindBugs**
- **Maven flash**
- **Help**
- **Maven IzPack**
- **Java Application**
- **Maven JAVANCSS**
- **Maven JAXB**
- **JUNITPP**
- **Kodo**
- **Maven Macker**
- **SDocBook**
- **Sourceforge**
- **Maven SpringGraph**
- **RPM Plugin**
- **Runtime Builder**
- **Strutsdoc**
- **Tasks**
- **Maven Transform**
- **Maven UberDist**
- **Maven Vignette**
- **WebSphere 4.0**
- **WebSphere 5 (5.0/5.1)**
- **Maven WebLogic**
- **Canoo WebTest**
- **Wiki**
- **Word to HTML**
- **XML Resume**
- **Maven DotUml**
- **Middlegen**
- **Maven News**

Good things about Maven

- Standardization
- Reuse
- Dependency management
- Build lifecycle management
- Large existing repository
- IDE aware
- One directory layout
- A single way to define dependencies
- Setting up a project is really fast
- Transitive dependencies
- Common build structure
- Use of remote repository
- Web site generation
- Build best practices enforcement
- Automated build of application
- Works well with distributed teams
- All artifacts are versioned and are stored in a repository
- Build process is standardized for all projects
- A lot of goals are available
- It provides quality project information with generated site
- Easy to learn and use
- Makes the build process much easier at the project level
- Promotes modular design of code

References

- **Maven Home**
<http://maven.apache.org/>
- **Maven Getting Started Guide**
<http://maven.apache.org/guides/getting-started/index.html>
- **Steps for creating a Maven-based Website**
http://www.javaworld.com/javaworld/jw-02-2006/jw-0227-maven_p.html
/
- **Maven Integration for Eclipse**
<http://m2eclipse.codehaus.org/>

Example and Demonstration

An example

<https://github.com/acherm/wikipediamatrix-bench>

Extracting Wikipedia tables into CSV (basic skeleton for testing/benchmarking solutions)

The screenshot shows the GitHub repository page for `wikipediamatrix-bench`. At the top, there are statistics: 8 commits, 1 branch, 0 releases, and 2 contributors. Below this, there are buttons for 'Branch: master', 'New pull request', 'Create new file', 'Upload files', 'Find File', and a prominent green 'Clone or download' button. The repository has one commit from `acherm` titled 'mistake in outputDirWikitext' made on 27 Nov 2018. There are also two other commits: one from `wikimatrix` and one from `README.md`. A file named `README.md` is currently selected, showing its content.

Wikipedia Matrix (benchmark)

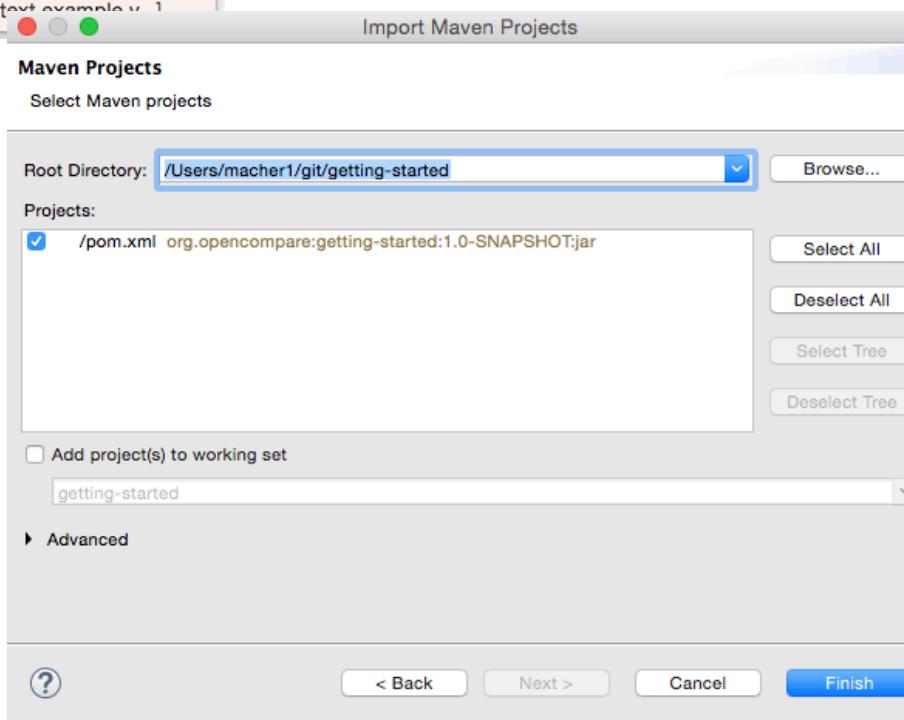
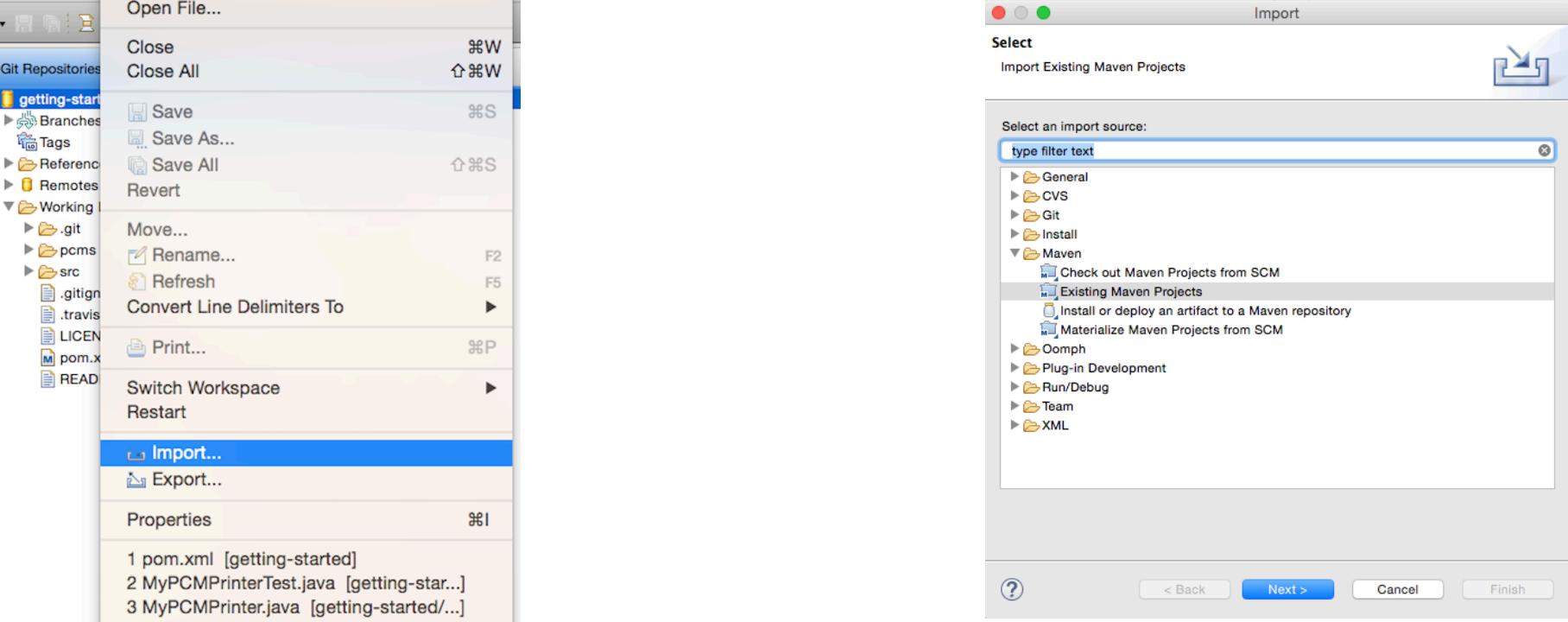
Extracting Wikipedia tables into CSV files (basic skeleton for testing/benchmarking solutions). Once the git is cloned:

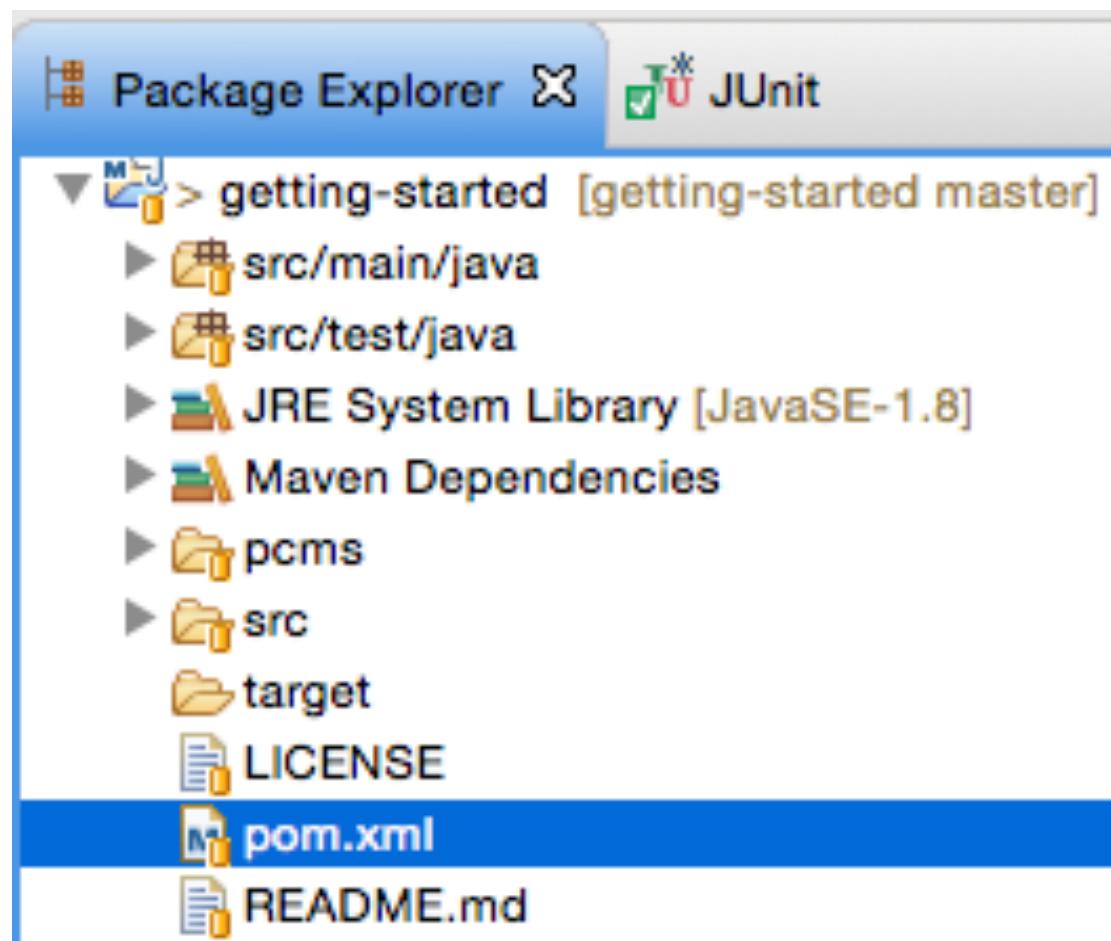
```
cd wikimatrix  
mvn test
```

We give 300+ Wikipedia URLs and the challenge is to:

- integrate the extractors' code (HTML and Wikitext)
- extract as many relevant tables as possible
- serialize the results into CSV files (within `output/html` and `output/wikitext`)

More details can be found in `BenchTest.java`. We are expecting to launch `mvn test` and the results will be in `output` folder





This XML file does not appear to have any style information associated with it. The document tree is

```
▼<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  ▼<parent>
    <artifactId>opencompare</artifactId>
    <groupId>org.opencompare</groupId>
    <version>0.5</version>
  </parent>
  <modelVersion>4.0.0</modelVersion>
  <artifactId>api-java</artifactId>
  ▼<dependencies>
    ▼<dependency>
      <groupId>com.opencsv</groupId>
      <artifactId>opencsv</artifactId>
      <version>3.3</version>
    </dependency>
    ▼<dependency>
      <groupId>org.scala-lang</groupId>
      <artifactId>scala-library</artifactId>
      <version>${scala.version}</version>
      <scope>test</scope>
    </dependency>
    ▼<dependency>
      <groupId>org.scalatest</groupId>
      <artifactId>scalatest_${scala.version.minor}</artifactId>
      <version>${scalatest.version}</version>
      <scope>test</scope>
    </dependency>
    ▼<dependency>
      <!-- jsoup HTML parser library @ http://jsoup.org/ -->
      <groupId>org.jsoup</groupId>
      <artifactId>jsoup</artifactId>
      <version>1.8.2</version>
    </dependency>
  </dependencies>
```

<http://search.maven.org/>

The Central Repository

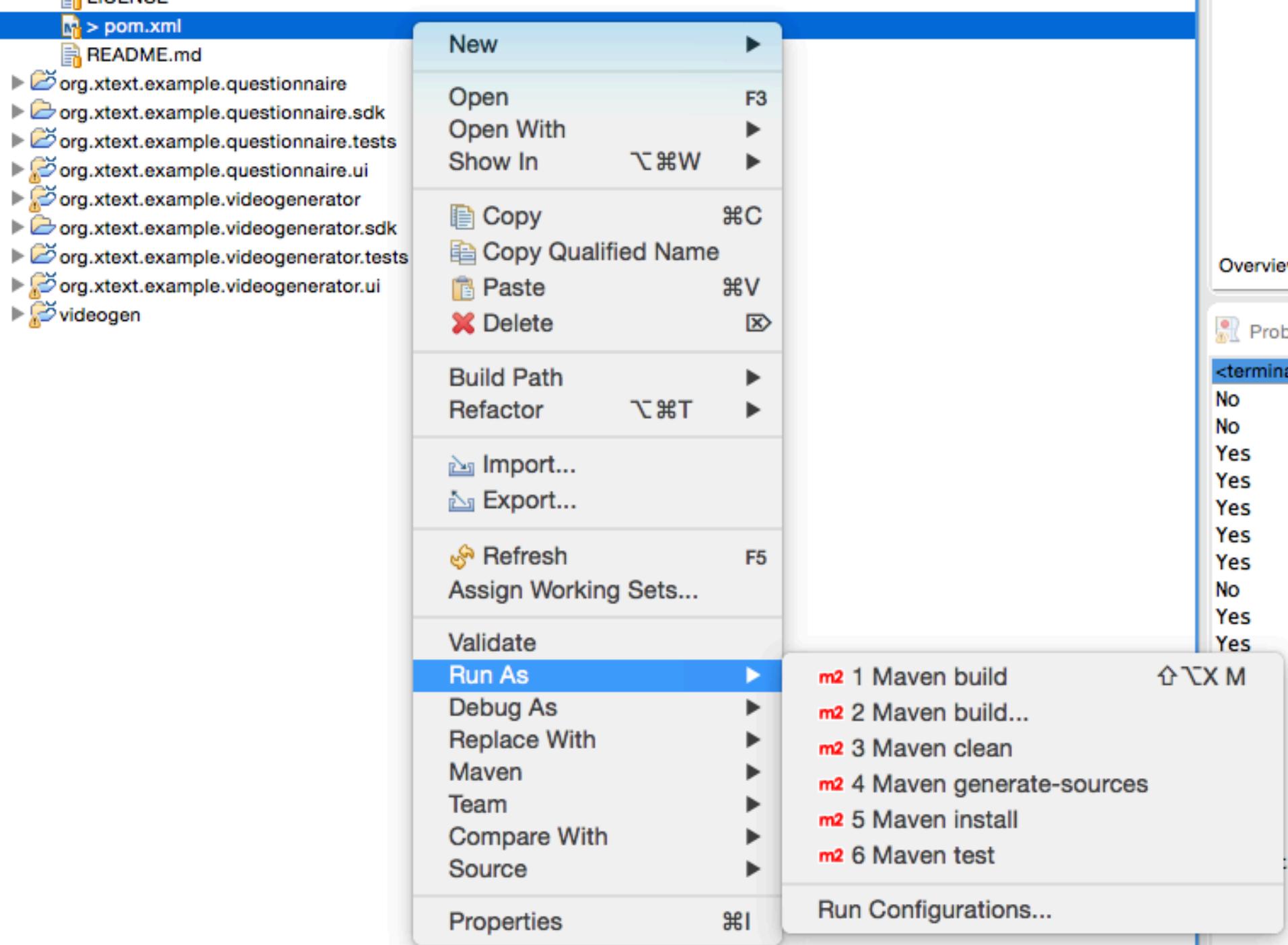
SEARCH | ADVANCED SEARCH | B

The Search Engine for The Central Repository

SEARCH

[New: About Central](#)

[Advanced Search](#) | [API Guide](#) | [Help](#)





Make



maven



GRUNT

The JavaScript Task Runner

Latest Version

- Stable: [v0.4.5](#) (npm)
- Development: [v0.4.6](#) (github)



Ads by [Bocoup](#).

[Free screencasts](#) about JavaScript, Flexbox, Node.js and more from the experts at Bocoup.

Why use a task runner?

In one word: automation. The less work you have to do when performing repetitive tasks like minification, compilation, unit testing, linting, etc, the easier your job becomes. After you've configured it through a [Gruntfile](#), a task runner can do most of that mundane work for you—and your team—with basically zero effort.

Why use Grunt?

The Grunt ecosystem is huge and it's growing every day. With literally hundreds of plugins to choose from, you can use Grunt to automate just about anything with a minimum of effort. If someone hasn't already built what you need, authoring and publishing your own Grunt plugin to npm is a breeze. See how to [get started](#).

<http://gruntjs.com/sample-gruntfile>



Bower

A package manager for the web

Travail actuel

- Installation (ce qu'il est attendu!)
 - Tests automatiques
 - **Avec Maven (indépendamment d'Eclipse/IntelliJ/VScode/...)**
 - Dans n'importe quel projet, **mvn test** doit fonctionner!
 - Automatisation sur n'importe quelle machine
 - Intégration continue

Today

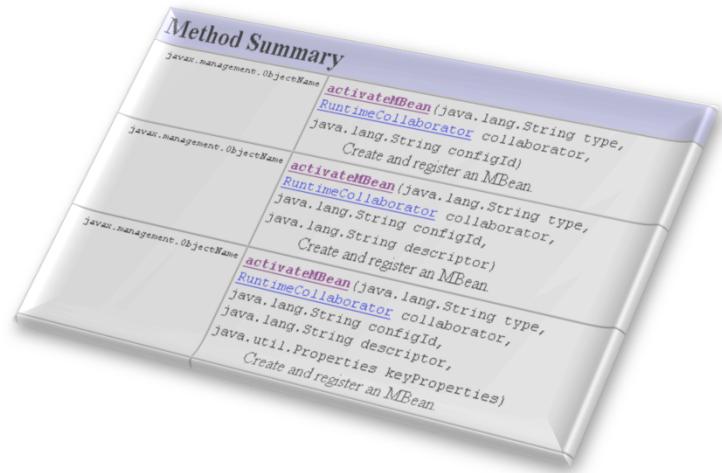
- Build System (maven)
- Manage your source code
- Refactoring
- Logging, Debugging and Test
- IDE (eg, Eclipse)
- Workflow: git, intégration continue, et tous les points ci-dessous

Manage your source code

Documentation

- Source code: one of the best artefact for documenting a project
- Javadoc (JDK)
 - Automatic **generation** of HTML documentation
 - Using comments in java files
- Syntax

```
/**  
 * This is a <b>doc</b> comment.  
 * @see java.lang.Object  
 * @todo fix {@underline this !}  
 */
```
- Includes
 - class hierarchy, interfaces, packages
 - detailed summary of class, interface, methods, attributes
- Note
 - Add doc generation to your favorite **compile chain**



Package javax.swing

Provides a set of "lightweight" (all-Java language) components that, to the maximum degree possible, work the same on all platforms.

See:

[Description](#)

Interface Summary

Action	The <code>Action</code> interface provides a useful extension to the <code>ActionListener</code> interface in cases where the same functionality may be accessed by several controllers.
BoundedRangeModel	Defines the data model used by components like Sliders and ProgressBars.
ButtonModel	State Model for buttons.
CellEditor	This interface defines the methods any general editor should be able to implement.
ComboBoxEditor	The editor component used for JComboBox components.
ComboBoxModel	A data model for a combo box.
DesktopManager	DesktopManager objects are owned by a JDesktopPane object.
Icon	A small fixed size picture, typically used to decorate components.
JComboBox.KeySelectionManager	The interface that defines a KeySelectionManager.
ListCellRenderer	Identifies components that can be used as "rubber stamps" to paint the cells in a JList.
ListModel	This interface defines the methods components like JList use to get the value of each cell in a list and the length of the list.
ListSelectionModel	This interface represents the current state of the selection for any of the components that display a list of values with stable indices.
MenuItem	Any component that can be placed into a menu should implement this interface.
MutableComboBoxModel	A mutable version of <code>ComboBoxModel</code> .
Renderer	Defines the requirements for an object responsible for "rendering" (displaying) a value.
RootPaneContainer	This interface is implemented by components that have a single JRootPane child: JDialog, JFrame, JWindow, JApplet, JInternalFrame.
Scollable	An interface that provides information to a scrolling container like JScrollPane.
ScrollPaneConstants	Constants used with the JScrollPane component.
SingleSelectionModel	A model that supports at most one indexed selection.
SpinnerModel	A model for a potentially unbounded sequence of object values.
SwingConstants	A collection of constants generally used for positioning and orienting components on the screen.
UIDefaults.ActiveValue	This class enables one to store an entry in the defaults table that's constructed each time it's looked up with one of the <code>getXXX(key)</code> methods.
UIDefaults.LazyValue	This class enables one to store an entry in the defaults table that isn't constructed until the first time it's looked up with one of the <code>getXXX(key)</code> methods.
WindowConstants	Constants used to control the window closing operation.

```
public class JFrame  
extends Frame  
implements WindowConstants, Accessible, RootPaneContainer
```

An extended version of `java.awt.Frame` that adds support for the JFC/Swing component architecture. You can find task-0

The `JFrame` class is slightly incompatible with `Frame`. Like all other JFC/Swing top-level containers, a `JFrame` contains a `JRootPane`, while the AWT `Frame` case. For example, to add a child to an AWT frame you'd write:

```
frame.add(child);
```

However using `JFrame` you need to add the child to the `JFrame`'s content pane instead:

```
frame.getContentPane().add(child);
```

The same is true for setting layout managers, removing components, listing children, and so on. All these methods should now throw a `java.awt.Container` exception. The default content pane will have a `BorderLayout` manager set on it.

update

```
public void update(Graphics g)
```

Just calls `paint(g)`. This method was overridden to prevent an unnecessary call to clear the background.

Overrides:

[update](#) in class [Container](#)

Parameters:

`g` - the Graphics context in which to `paint`

See Also:

[Component.update\(Graphics\)](#)



Kornel Kisielewicz @epyoncf

12 Aug

ProTip: "://" is the speedup operator. Use // before the statement you want to speed up. Works in C++, Java and a few others!

Retweeted by Mathieu Acher

Collapse

Reply

Retweeted

Favorite

More

1,253

RETWEETS

295

FAVORITES



12:31 AM - 12 Aug 13 · Details

Coding Conventions

- Rules on the coding style :
 - Apache, Oracle and others template
 - e.g.
<http://www.oracle.com/technetwork/java/codeconv-138413.html>
 - <http://geosoft.no/development/javastyle.html>
- Verification tools
 - CheckStyle, PMD, JackPot, Spoon Vsuite...
 - Some integrated into IDEs

Why Coding Standards are Important?

- Lead to greater **consistency** within your code and the code of your teammates
- Easier to **understand**
- Easier to **develop**
- Easier to **maintain**
- Reduces overall cost of application

Example

8. Private class variables should have underscore suffix.

```
class Person
{
    private String name_;
    ...
}
```

Apart from its name and its type, the scope of a variable is its most higher significance than method variables, and should be treated w

A side effect of the underscore naming convention is that it nicely i

```
void setName(String name)
{
    name_ = name;
}
```

Code Style and Code Conventions: many languages (Java, XML, JavaScript, HTML, CSS, etc.)

An example:

<https://maven.apache.org/developers/conventions/code.html>

The following is the recommended ordering for all Maven POM files:

```
1. <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
2.   <modelVersion/>
3.
4.   <parent/>
5.
6.   <groupId/>
7.   <artifactId/>
8.   <version/>
9.   <packaging/>
10.
11.  <name/>
12.  <description/>
13.  <url/>
14.  <inceptionYear/>
15.  <organization/>
16.  <licenses/>
17.
18.  <developers/>
19.  <contributors/>
20.
21.  <mailingLists/>
22.
23.  <prerequisites/>
24.
25.  <modules/>
26.
27.  <scm/>
28.  <issueManagement/>
29.  <ciManagement/>
30.  <distributionManagement/>
31.
32.  <properties/>
33.
34.  <dependencyManagement/>
35.  <dependencies/>
36.
37.  <repositories/>
38.  <pluginRepositories/>
39.
```

<https://maven.apache.org/developers/conventions/code.html>

<https://maven.apache.org/developers/conventions/code.html>

Generic Code Style And Convention

All working files (java, xml, others) should respect the following conventions:

- **License Header:** Always add the current [ASF license header](#) in all versionned files.
- **Trailing Whitespaces:** Remove all trailing whitespaces. If you are an Eclipse user, you could use the [Anyedit Eclipse Plugin](#).

and the following style:

- **Indentation:** Never use tabs!
- **Line wrapping:** Always use a 120-column line width.

Note: The specific styles and conventions, listed in the next sections, could override these generic rules.

Java

Java Code Style

The Maven style for Java is mainly:

- **White space:** One space after control statements and between arguments (i.e. `if (foo)` instead of `if(foo)`), `myFunc(foo, bar, baz)` instead of `myFunc(foo,bar,baz)`). No spaces after methods names (i.e. `void myMethod()`, `myMethod("foo")`)
- **Indentation:** Always use 4 space indents and **never** use tabs!
- **Blocks:** Always enclose with a new line brace.
- **Line wrapping:** Always use a 120-column line width for Java code and Javadoc.

Tools to Improve your Source code

- Formatting tools
 - Indenters (Jindent), beautifiers, stylers (JavaStyle), ...
- Code conventions/styles:
 - Eg Checkstyle
 - Exists as a Maven plugin (<https://maven.apache.org/plugins/maven-checkstyle-plugin/>)
- Quality report tools: code metrics
 - Number of Non Comment Code Source, Number of packages, Cyclomatic numbers, ...
 - JavaNCCS, Eclipse Metrics ...

Today

- Build System (maven)
- Manage your source code
- **Refactoring**
- Logging, Debugging and Test
- IDE (eg, Eclipse)
- Workflow: git, intégration continue, et tous les points ci-dessous

Refactoring

What's Code Refactoring?

“A series of *small* steps, each of which changes the program’s *internal structure* without changing its *external behavior*”



Martin Fowler

Example

Which code segment is easier to read?

Sample 1:

```
if (markT>=0 && markT<=25 && markL>=0 && markL<=25) {  
    float markAvg = (markT + markL)/2;  
    System.out.println("Your mark: " + markAvg);  
}
```

Sample 2:

```
if (isValid(markT) && isValid(markL)) {  
    float markAvg = (markT + markL)/2;  
    System.out.println("Your mark: " + mark);  
}
```

Why do we Refactor?

- Improves the design of our software
 - Design pattern!
- Minimizes technical debt
- Keep development at speed
- To make the software easier to understand
- To help find bugs
- To “Fix broken windows”

Non exhaustive (code smell)

(and not necessarily smells in all situations)

- Duplicated code
- Feature Envy
- Inappropriate Intimacy
- Comments
- Long Method
- Long Parameter List
- Switch Statements
- Improper Naming

Code Smell examples (1)

```
public void display(String[] names) {  
    System.out.println("-----");  
    for(int i=0; i<names.length; i++){  
        System.out.println(" " + " " + names[i]);  
    }  
    System.out.println("-----");  
}  
  
public void listMember(String[] names) {  
    System.out.println("List all member: ");  
    System.out.println("-----");  
    for(int i=0; i<names.length; i++){  
        System.out.println(" " + " " + names[i]);  
    }  
    System.out.println("-----");  
}
```

Duplicated code

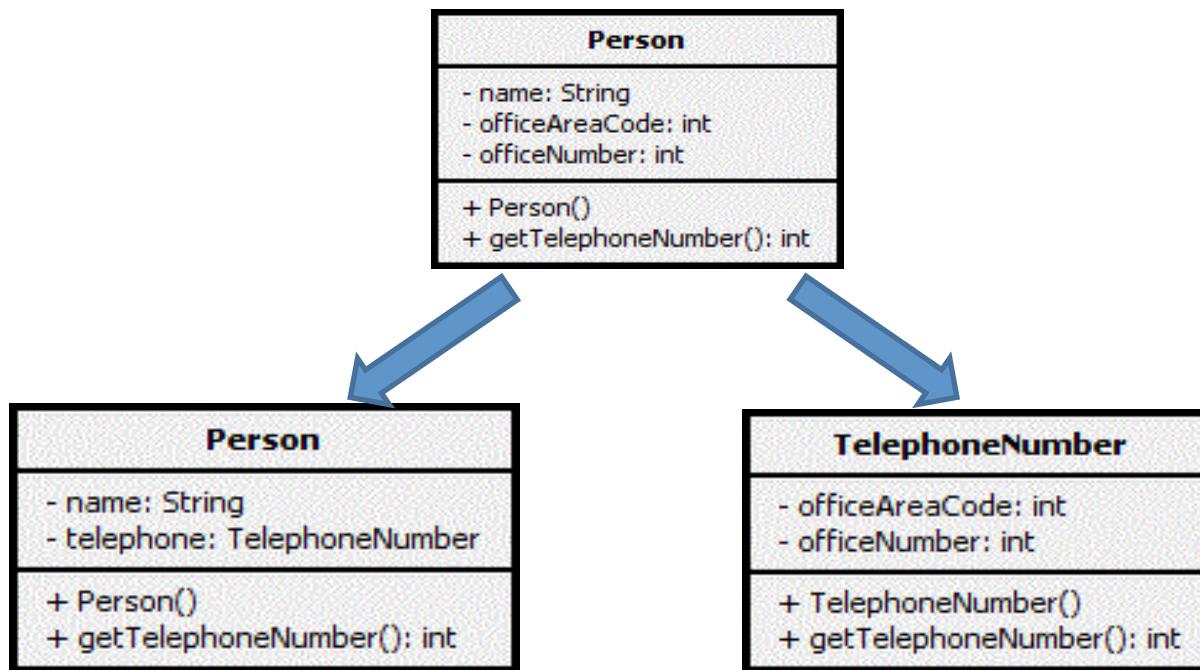
Code Smell examples (2)

```
public String formatStudent( int id,  
                           String name,  
                           Date dob,  
                           String province,  
                           String address,  
                           String phone ) {  
  
    //TODO:  
    return null;  
}
```

Long list of parameters

Improving design

- Move Method or Move Field – move to a more appropriate Class or source file
- Rename Method or Rename Field – changing the name into a new one that better reveals its purpose
 - Pull Up – in OOP, move to a superclass
 - Push Down – in OOP, move to a subclass



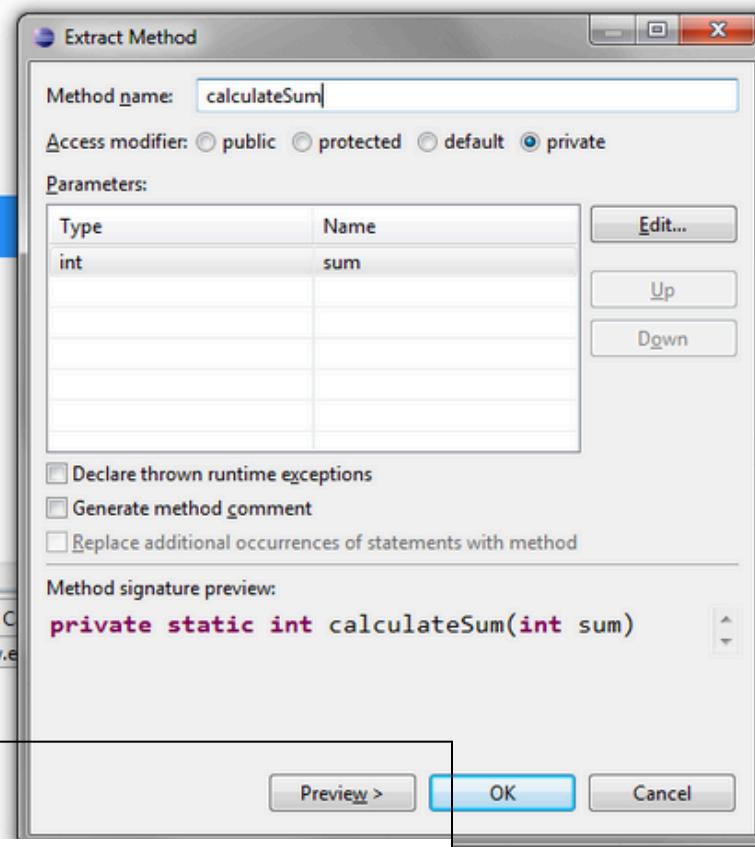
How do we Refactor?

- Manual Refactoring
 - Code Smells
- Automated/Assisted Refactoring
 - Refactoring by hand is time consuming and prone to error
 - Tools (IDE)
- In either case, **test your changes**

```
package de.vogella.eclipse.ide.first;

public class MyFirstClass {

    public static void main(String[] args) {
        System.out.println("Hello Eclipse!");
        int sum = 0;
        for (int i = 0; i <= 100; i++) {
            sum += i;
        }
        System.out.println(sum);
    }
}
```



```
package de.vogella.eclipse.ide.first;

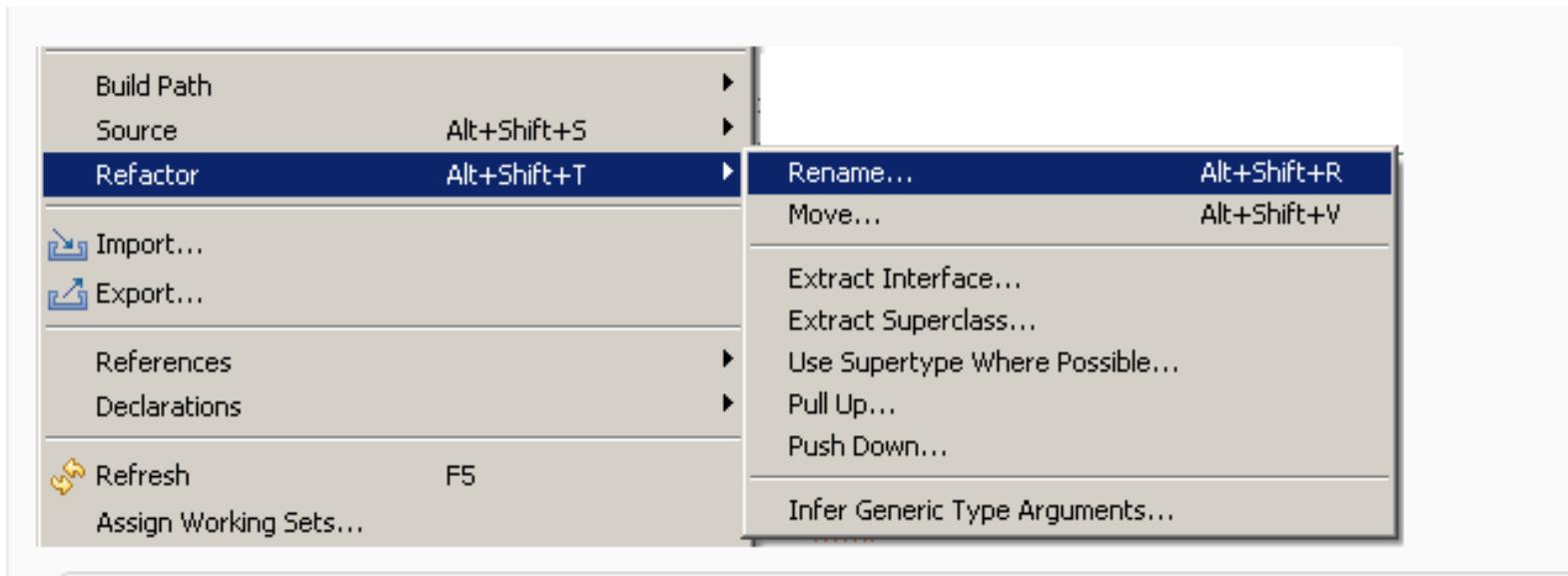
public class MyFirstClass {

    public static void main(String[] args) {
        System.out.println("Hello Eclipse!");
        int sum = 0;
        sum = calculateSum(sum);
        System.out.println(sum);
    }

    private static int calculateSum(int sum) {
        for (int i = 0; i <= 100; i++) {
            sum += i;
        }
        return sum;
    }
}
```

Typical refactoring patterns

- Rename variable / class / method / member
- Extract method
- Extract constant
- Extract interface
- Encapsulate field



You have constructors on subclasses with mostly identical bodies.

Create a superclass constructor; call this from the subclass methods.

Pull Up Constructor Body

```
class Manager extends Employee...
    public Manager (String name, String id, int grade) {
        _name = name;
        _id = id;
        _grade = grade;
    }
```

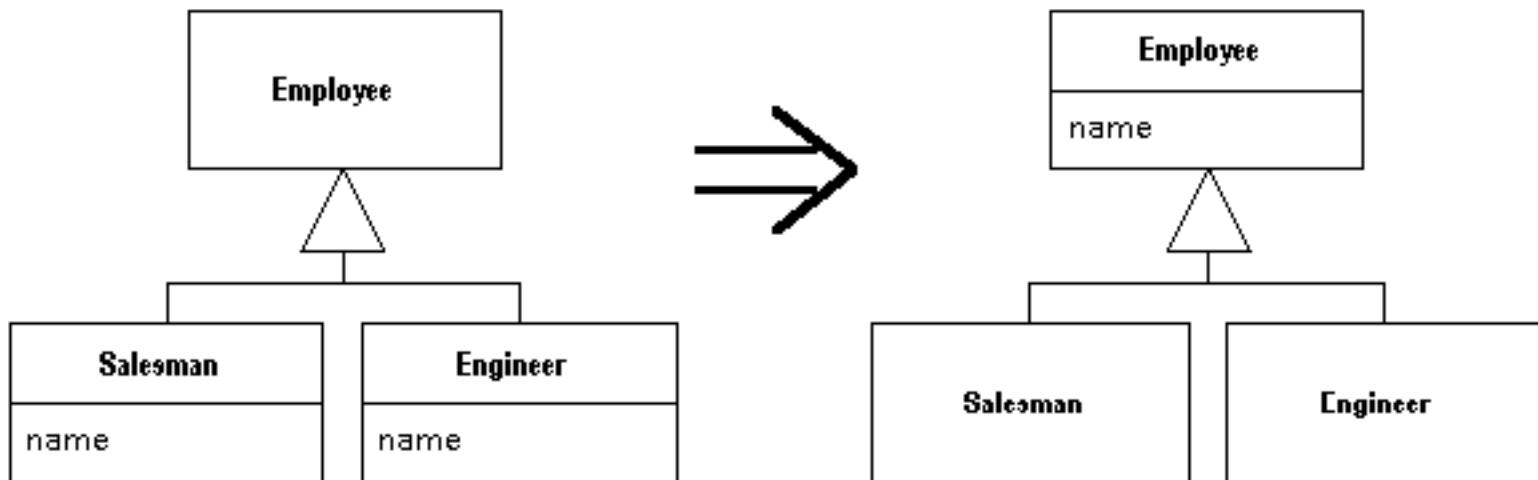
```
public Manager (String name, String id, int grade) {
    super (name, id);
    _grade = grade;
}
```

You

Create

Two subclasses have the same field.

Move the field to the superclass.



You have a complicated expression.

Put the result of the expression, or parts of the expression, in a temporary variable with a name that explains the purpose.

```
if ( (platform.toUpperCase().indexOf("MAC") > -1) &&
    (browser.toUpperCase().indexOf("IE") > -1) &&
    wasInitialized() && resize > 0 )
{
    // do something
}

final boolean isMacOs      = platform.toUpperCase().indexOf("MAC") > -1;
final boolean isIEBrowser = browser.toUpperCase().indexOf("IE")  > -1;
final boolean wasResized  = resize > 0;

if (isMacOs && isIEBrowser && wasInitialized() && wasResized)
{
    // do something
}
```

CONTROLLABILITY

ability to manipulate the software's input as well as to place this software into a particular state

OBSERVABILITY

deals with the possibility to observe the outputs and state changes that

Inputs



Outputs

TESTABILITY

degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met.

Controllability + Observability

Inputs



Outputs

Conclusion

- How to improve Testability?
 - Refactoring, Design patterns
 - Separation of concerns, Modularity, Abstractions
- Logging
- Debugging
- Testing



Whenever you are tempted to type something into a print statement or a debugger expression, write it as a test instead.

- Logging
 - Manual observation
 - (Usually) manual control on input values
 - Pre-defined exploration of values
- Debugging
 - Manual observation
 - (Usually) manual control on input values
 - Interactive, fine-grained exploration of values
- Testing
 - Automated control and observation (assertions)
 - More amenable to re-executing on different inputs
 - Not to understand, but to verify some properties

Today

- Build System (maven)
- Manage your source code
- Refactoring
- Logging, Debugging and Test
- IDE (eg, Eclipse)
- Workflow: git, intégration continue, et tous les points ci-dessous

Document, refactor... Execute your tests... Debug.. Write test..

Documenting

And so on!

Refactoring

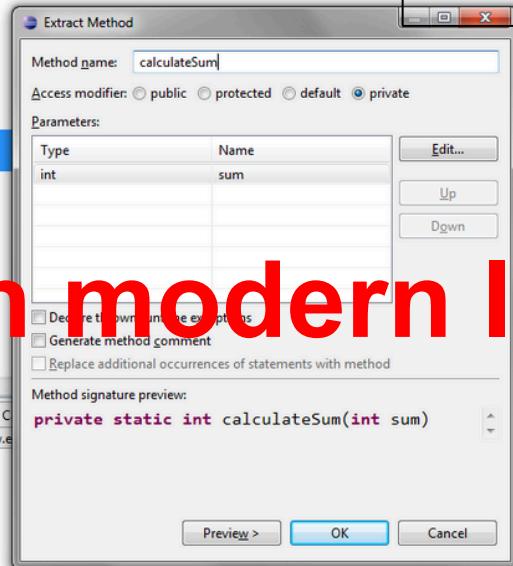
Debugging

Testing

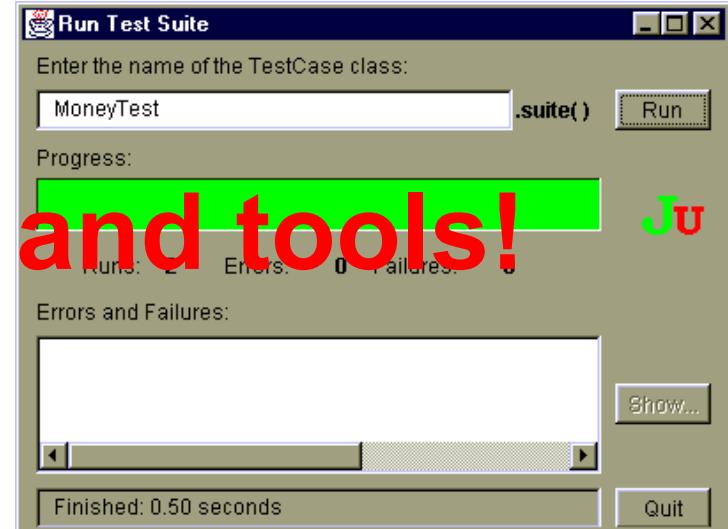
```
package de.vogella.eclipse.ide.first;

public class MyFirstClass {

    public static void main(String[] args) {
        System.out.println("Hello Eclipse!");
        int sum = 0;
        for (int i = 0; i <= 100; i++) {
            sum += i;
        }
        System.out.println(sum);
    }
}
```



With modern IDE and tools!

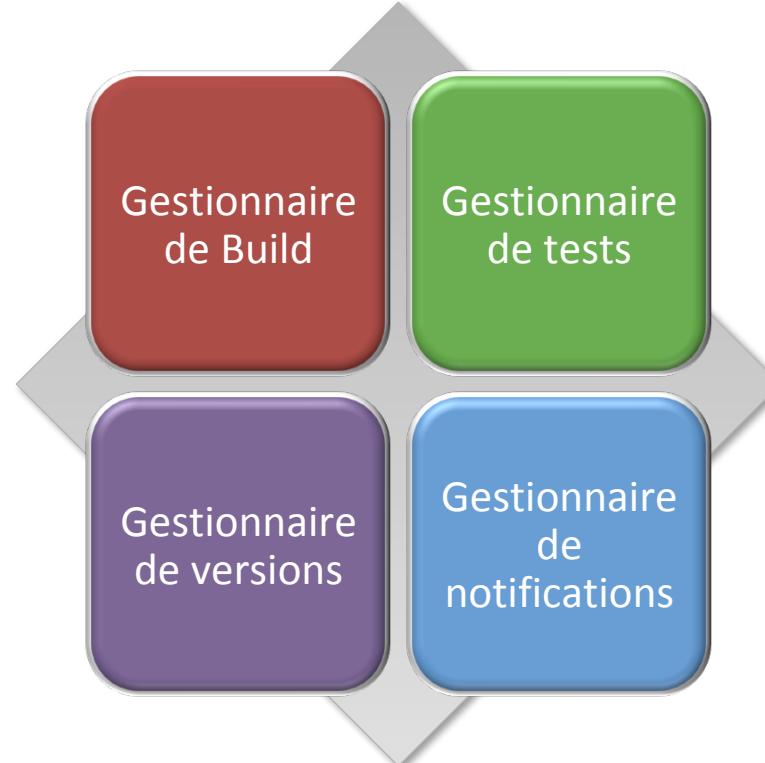


Continuous Integration

« L'intégration continue est un ensemble de pratiques utilisées en génie logiciel consistant à vérifier à chaque **modification** de code source que le résultat des modifications ne produit pas de **régression** dans l'application développée. »

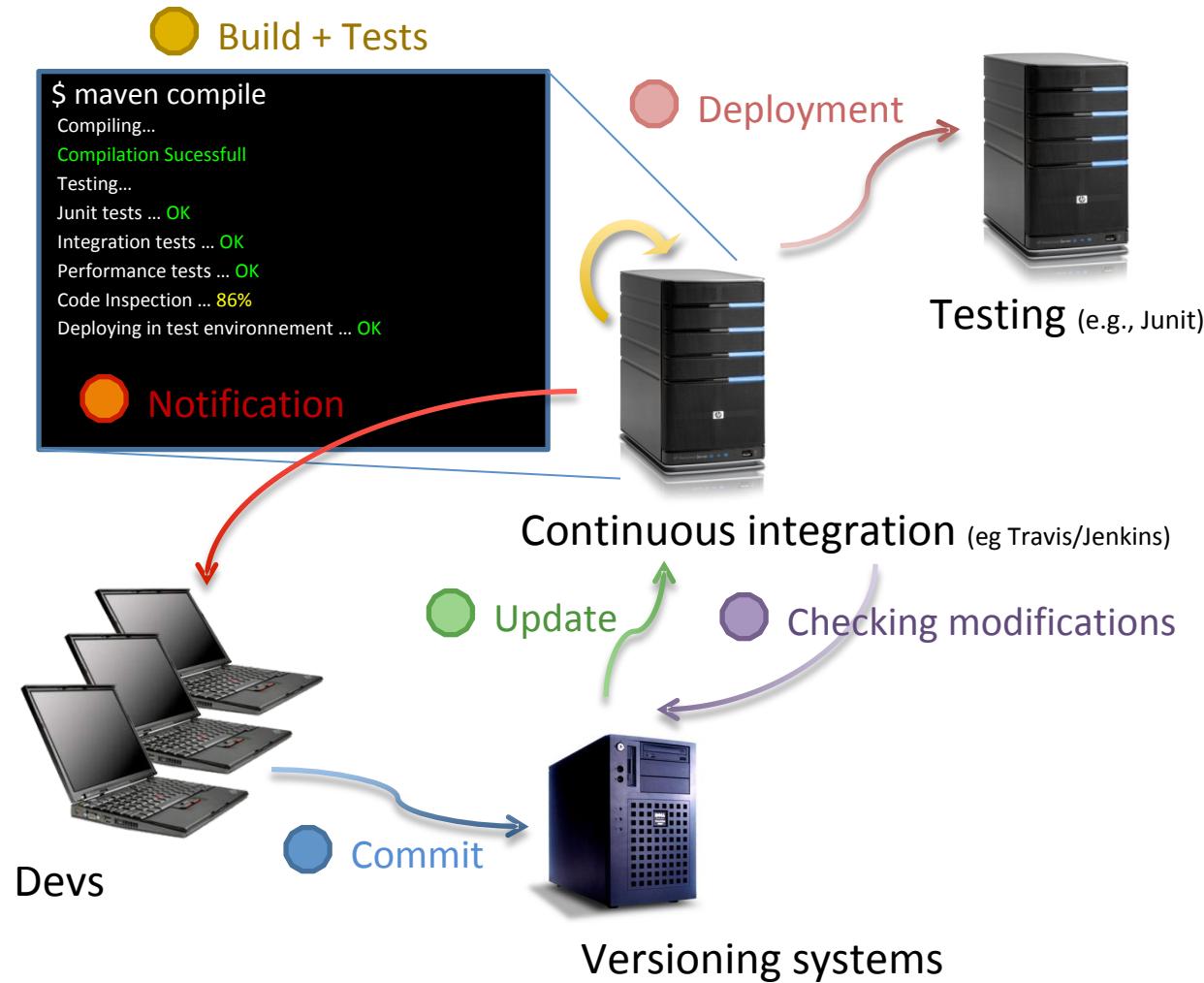
maven

 **git**



JUnit.org

Usage





OpenCompare / getting-started

Branch: **master** ▾

getting-started / .travis.yml



gbecan 10 days ago add .gitignore and .travis.yml file

1 contributor

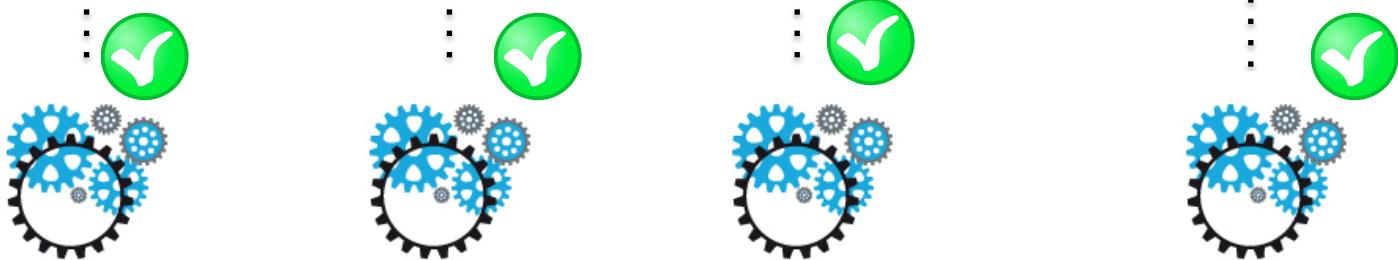
6 lines (4 sloc) | 62 Bytes

```
1 language: java
2 jdk:
3   - oraclejdk8
4
5 script: mvn clean install
```

Travail actuel

- Installation (ce qu'il est attendu!)
 - Tests automatiques
 - Avec Maven (indépendamment d'Eclipse/IntelliJ/VScode/...)
 - **Intégration continue**

SP (sprints; implémentation)



Execute the tests before/after each commit
Don't break (no regression)
Continuous validation

Impacts

- Use/experiment with all these tools
 - IDE in general (Eclipse, IntelliJ, etc.) and all services...
 - Debugging
 - Refactoring
 - Testing
 - Documentation
 - Maven
 - Versioning systems (git)
- You will have to in your professional career!