software assessment
Boeing B-29 „Superfortress“
disassemble
disassemble

run

test and compare
software assessment

tudorgirba.com
development
assessment

data → knowledge
data → analyses → knowledge
reverse engineering
data → analyses → models → knowledge
assessment

data → analyses → models → knowledge
interview during demo

Object-Oriented Reengineering

Patterns

Serge Demeyer, Stéphane Ducasse, Oscar Nierstrasz

chat with maintainers
read all code in one hour
/**
 * We hang our heads in shame. There are still bugs in ArgoUML
 * and/or GEF that cause corruptions in the model.
 * Before a save takes place we repair the model in order to
 * be as certain as possible that the saved file will reload.
 * TODO: Split into small inner classes for each fix.
 *
 * @return A text that explains what is repaired.
 */
updateTypeAccordingToEntities

"-- ugly code, will change once we move to CollectiveBehavior --"

| common wantedType class |
common := self commonMetaDescription.
wantedType := (common name, 'Group') asSymbol.
self metaDescription name == wantedType ifTrue: [ ^self ].
class := AbstractGroup allSubclasses
detect: [ :each | each asMetaDescription name == wantedType ]
ifNone: [ ^self changeTypeToDefaultType ].
self changeTypeTo: class.
blink

By the author of The Tipping Point

* The Power of Thinking Without Thinking

Malcolm Gladwell

Sources of Power

How People Make Decisions

Gary Klein

predict you'll be hearing a lot more about intuitive decision-making.
—Thomas Petzinger, Jr., Wall Street Journal
public class Library {
    List books;
    public Library() {...}
    public void addBook(Book b) {...}
    public void removeBook(Book b) {...}
    private boolean hasBook(Book b) {...}
    protected List getBooks() {...}
    protected void setBooks(List books) {...}
    public boolean equals(...) {...}
}

NOM = ?
public class Library {
    List books;
    public Library() {...}
    public void addBook(Book b) {...}
    public void removeBook(Book b) {...}
    private boolean hasBook(Book b) {...}
    protected List getBooks() {...}
    protected void setBooks(List books) {...}
    public boolean equals(…) {...}
}
public class Library {
    List books;
    public Library() {...}
    public void addBook(Book b) {...}
    public void removeBook(Book b) {...}
    private boolean hasBook(Book b) {...}
    protected List getBooks() {...}
    protected void setBooks(List books) {...}
    public boolean equals(…) {...}
}

NOM = 6
public class Library {
    List books;
    public Library() {…}
    public void addBook(Book b) {…}
    public void removeBook(Book b) {…}
    private boolean hasBook(Book b) {…}
    protected List getBooks() {…}
    protected void setBooks(List books) {…}
    public boolean equals(…) {…}
}
public class Library {
    List books;
    public Library() {…}
    public void addBook(Book b) {…}
    public void removeBook(Book b) {…}
    private boolean hasBook(Book b) {…}
    protected List getBooks() {…}
    protected void setBooks(List books) {…}
    public boolean equals(…) {…}
}
public class Library {
    List books;
    public Library() {…}
    public void addBook(Book b) {…}
    public void removeBook(Book b) {…}
    private boolean hasBook(Book b) {…}
    protected List getBooks() {…}
    protected void setBooks(List books) {…}
    public boolean equals(…) {…}
}

NOM = 7 6 4 3 2
public class Library {
    List books;
    public Library() {...}
    public void addBook(Book b) {...}
    public void removeBook(Book b) {...}
    private boolean hasBook(Book b) {...}
    protected List getBooks() {...}
    protected void setBooks(List books) {...}
    public boolean equals(…) {...}
}

NOM = 7, 6, 4, 3, 2 ?
public class Library {
    List books;
    public Library() {...}
    public void addBook(Book b) {...}
    public void removeBook(Book b) {...}
    private boolean hasBook(Book b) {...}
    protected List getBooks() {...}
    protected void setBooks(List books) {...}
    public boolean equals(…) {...}
}

NOM = 7, 6, 4, 3, 2 ?

your responsibility
data → analyses → models → knowledge
data -> importers -> models -> analyses
classes select: #isAnnotated

McCabe = 21
LOC = 753,000
classes select: #isGod

McCabe = 21
LOC = 753,000
classes select: #isGod

McCabe = 21  LOC = 753,000
classes select: #isGod

McCabe = 21

LOC = 753,000
classes select: #isGod

McCabe = 21
LOC = 753,000
classes select: #isGod

McCabe = 21
LOC = 753,000

Tuesday, October 25, 11
What is this made of?
$\textbf{Mondrian Easel}$

**Script**

```plaintext
view interaction menu: #mooseMenu.
view shape rectangle
  height: #numberOfMethods;
  width: #numberOfAttributes;
  linearFillColor: #numberOfLinesOfCode within: classGroup.
view nodes: classGroup.
view edgesFrom: #superclass.
view treeLayout
```

**Variable**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>classGroup</td>
<td>All model...Classes)</td>
</tr>
</tbody>
</table>
What is this made of?
composer tabulator with: [:t |
  t row: [:r | r column: #namespaces;
          column: #classes; column: #methods];
  row: #details.
  t transmit to: #namespaces; andShow: [:a |
    a tree
    title: 'Namespaces';
    display: [:m | m allNamespaces select: #isRoot ];
    children: #childScopes;
    format: #name ].
  t transmit from: #namespaces; to: #classes; andShow: [:a |
    a list
    title: 'Classes';
    display: [:n | n classes ];
    format: #name].
  t transmit from: #classes; to: #methods; andShow: [:a |
    a list
    title: 'Methods';
    display: [:c | c methods ];
    format: #name].
  t transmit from: #methods; to: #details; andShow: [:a |
    a text
    display: #formattedSourceText ]
].
composer openOn: model
Moose is a platform for software and data analysis.

It is an open source project since 1996. It is supported by several research groups around the world, and it is increasingly adopted in industrial projects.

- Download 4.6
- The Moose Book
the book
that shows
the outside
the inside and
the philosophy of
the Moose platform

by Tudor Girba

About
This book offers an overview of the Moose platform for software and data analysis. More specifically it covers version 4.

Currently, the book is in a preliminary shape, with a number of parts still under writing.

Feedback is more than appreciated. Please contact me, or leave a comment on this site.

themoosebook.org
development

assessment
development

assessment

- explicit
- tailored
- educated
software assessment
humane
essessment

tudorgirba.com