A Constraint Programming Approach to Conflict-aware Optimal Scheduling of Prioritized Code Clone Refactoring

Minhaz Fahim Zibran
PhD Student

Chanchal K. Roy
Assistant Professor

UNIVERSITY OF SASKATCHEWAN

Background: effect of cloning

```
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ......
    writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

Background: effect of cloning

```java
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ......
    writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```java
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ......
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

Background: effect of cloning

```
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ......
    writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ......
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ....
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

Background: effect of cloning

```java
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ......
    writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```java
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ......
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```java
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ....
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```java
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ....
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

Background: effect of cloning

```
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ......
    writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ......
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ....
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ....
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ....
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

Background: effect of cloning

```java
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ......
    writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```java
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ......
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```java
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ....
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```java
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ....
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```java
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ....
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

Background: effect of cloning

```
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ......
    writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

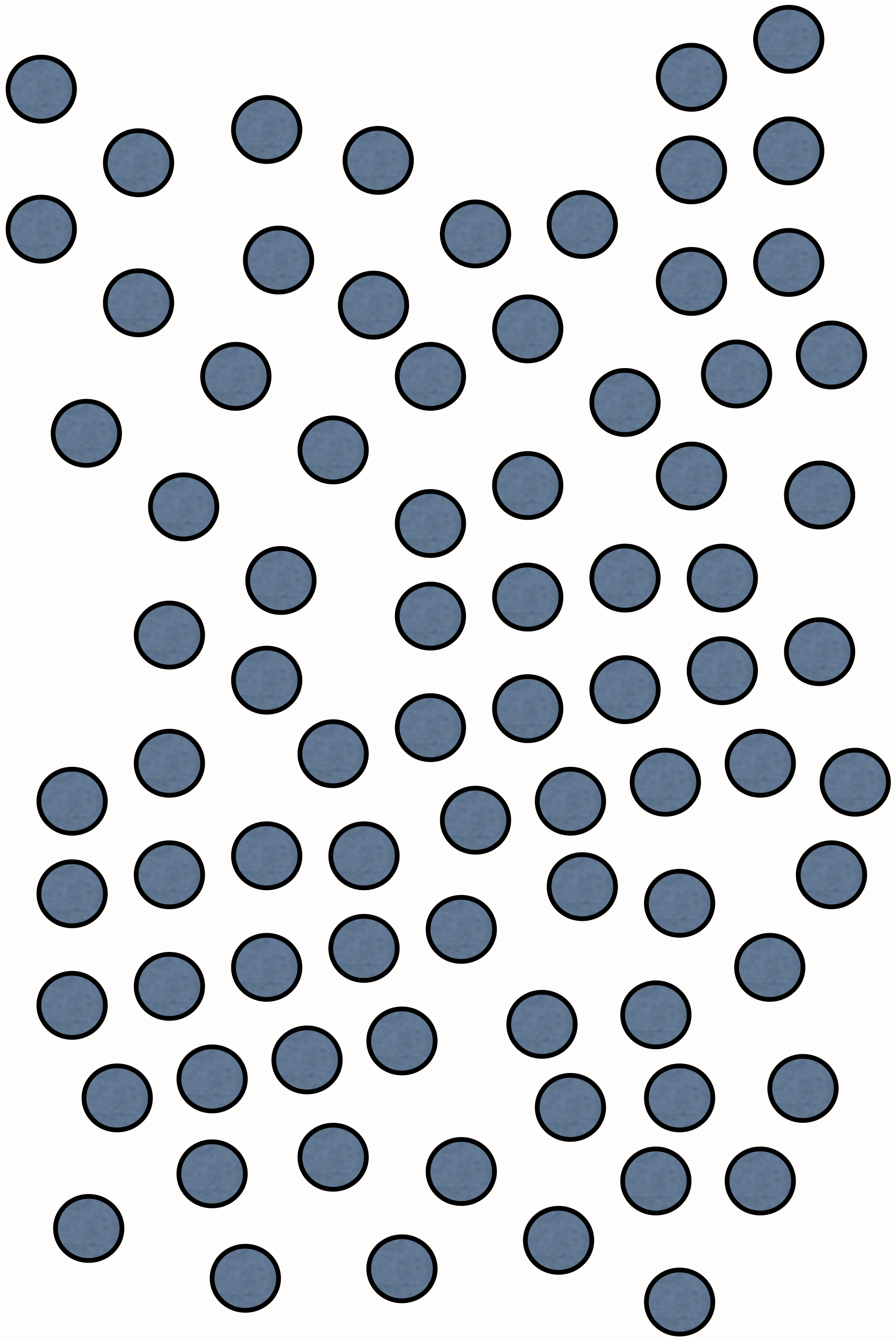
```
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ......
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ....
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

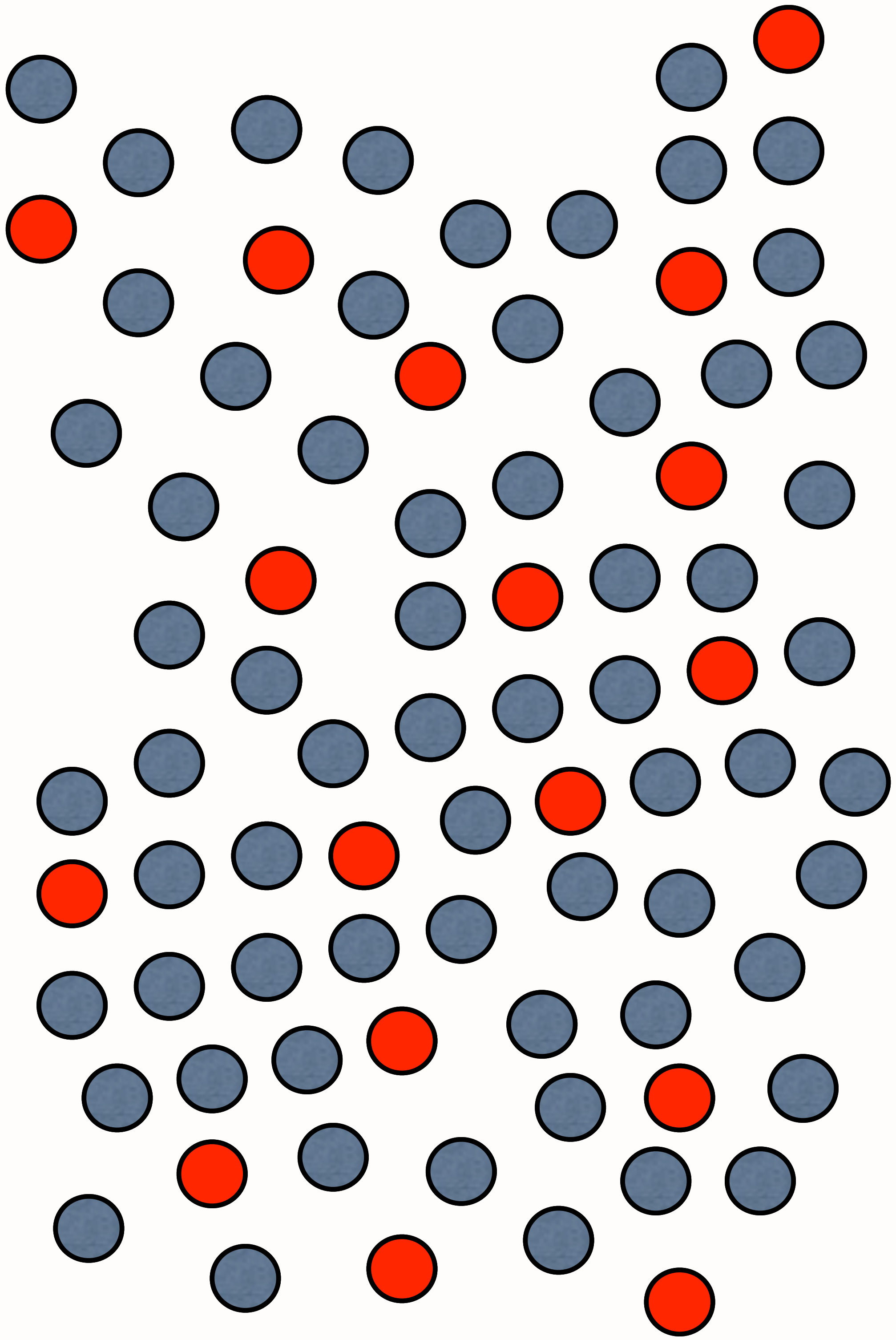
```
PrintWriter writer = null;
try {
    writer = new PrintWriter(new FileWriter(file));
    writer.write(...);
    //........ more statements ....
    //writer.close();
} catch (IOException e) {
    e.printStackTrace();
}
```

```
PrintWriter writer = null;
try {
    writer = new PrintWriter( ew FileWrite ile));
    writer.write(...);
    //........ more stat ents ....
    //writer.close();
} catch (IOException e) {
    e.printStackTrace()
}
```

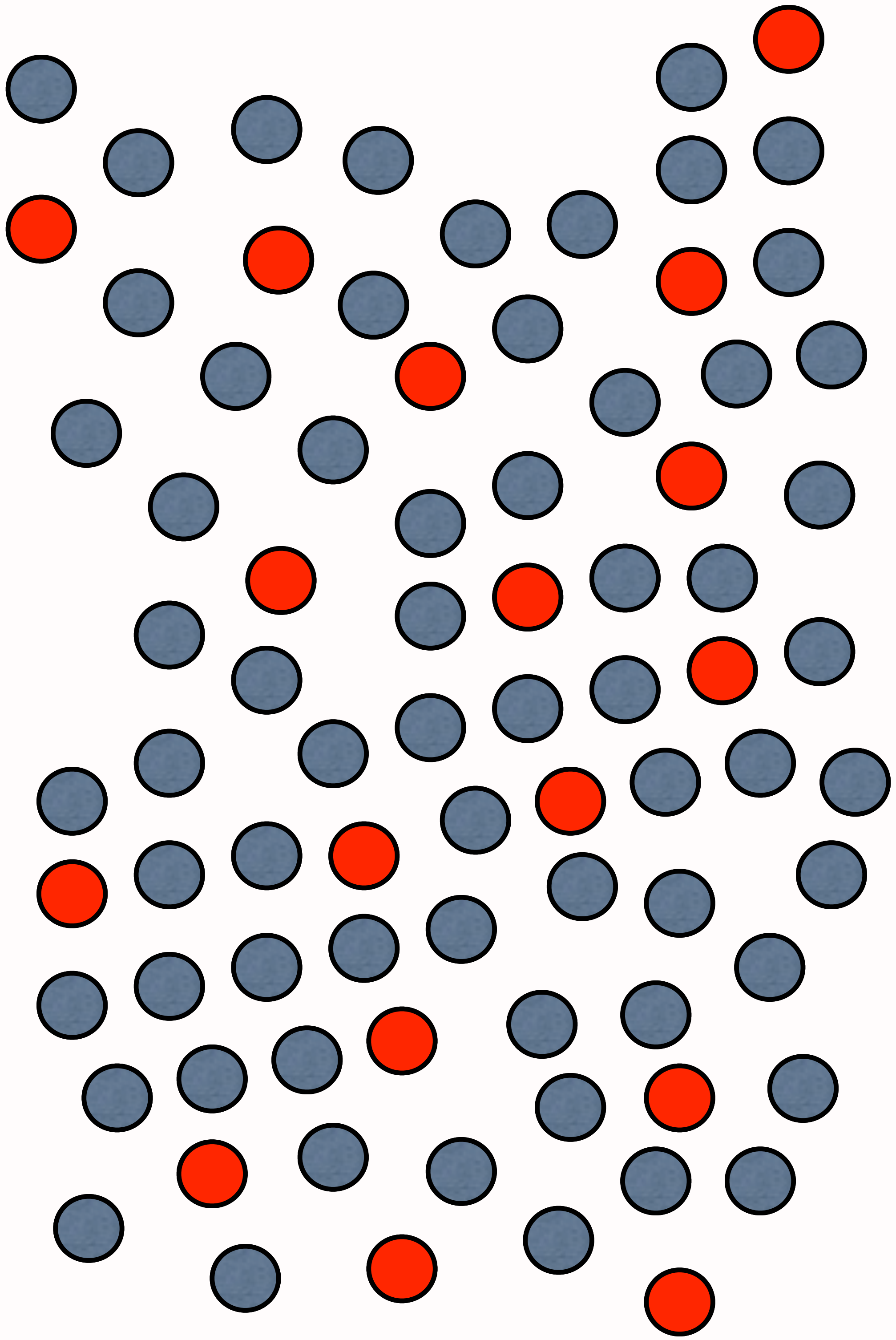
The Problem

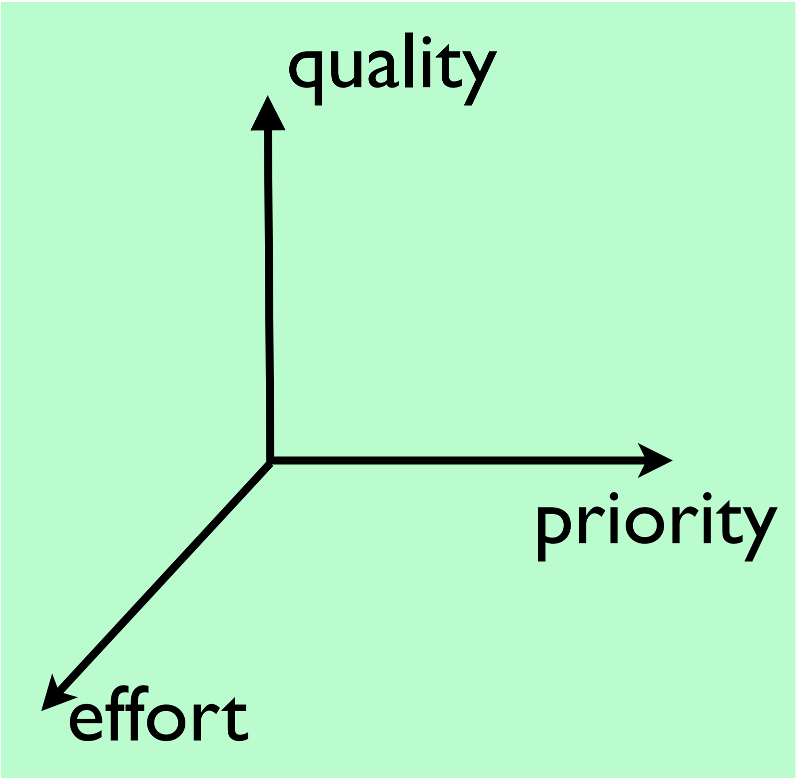


The Problem

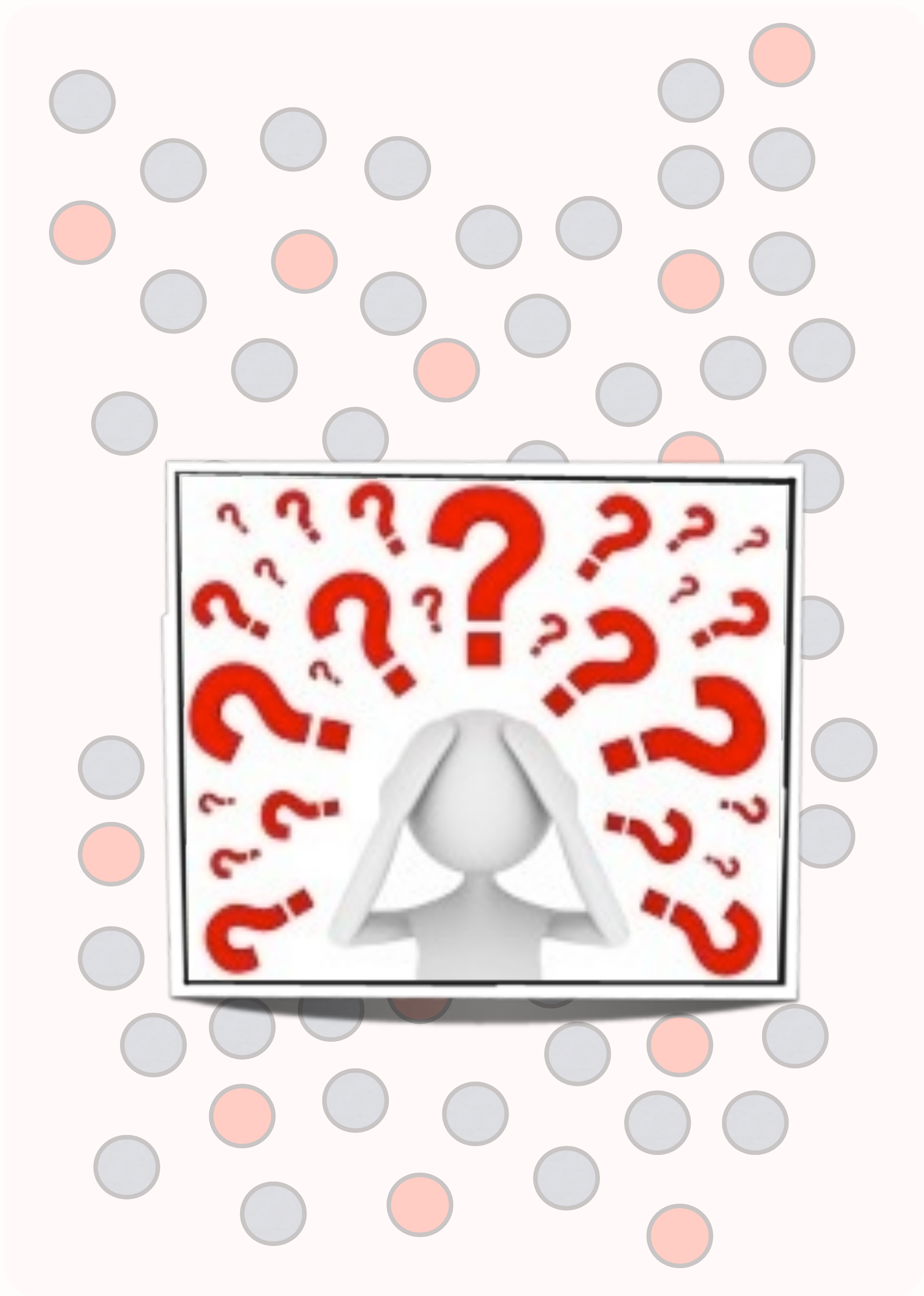


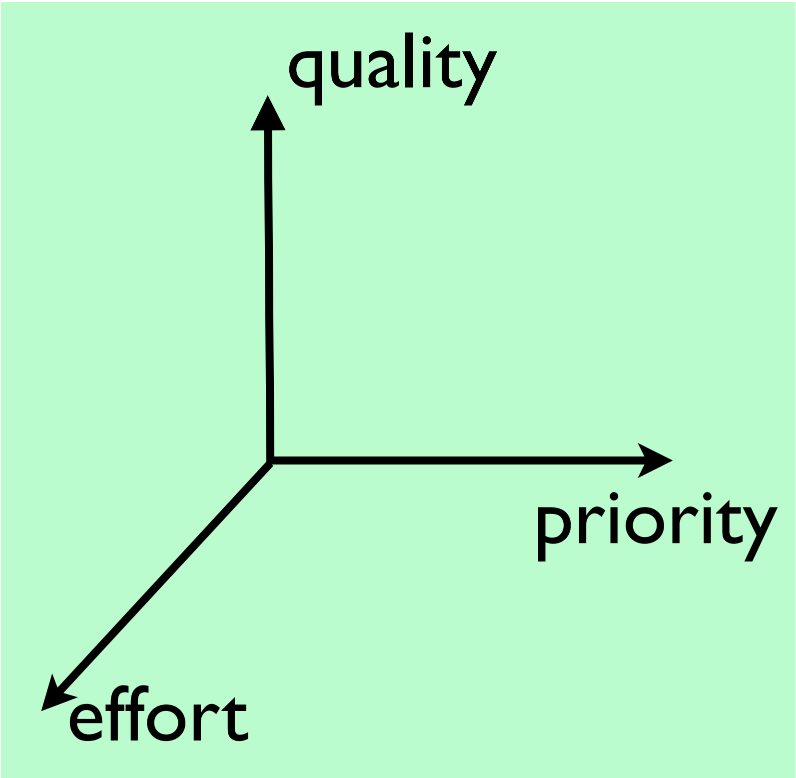
The Problem



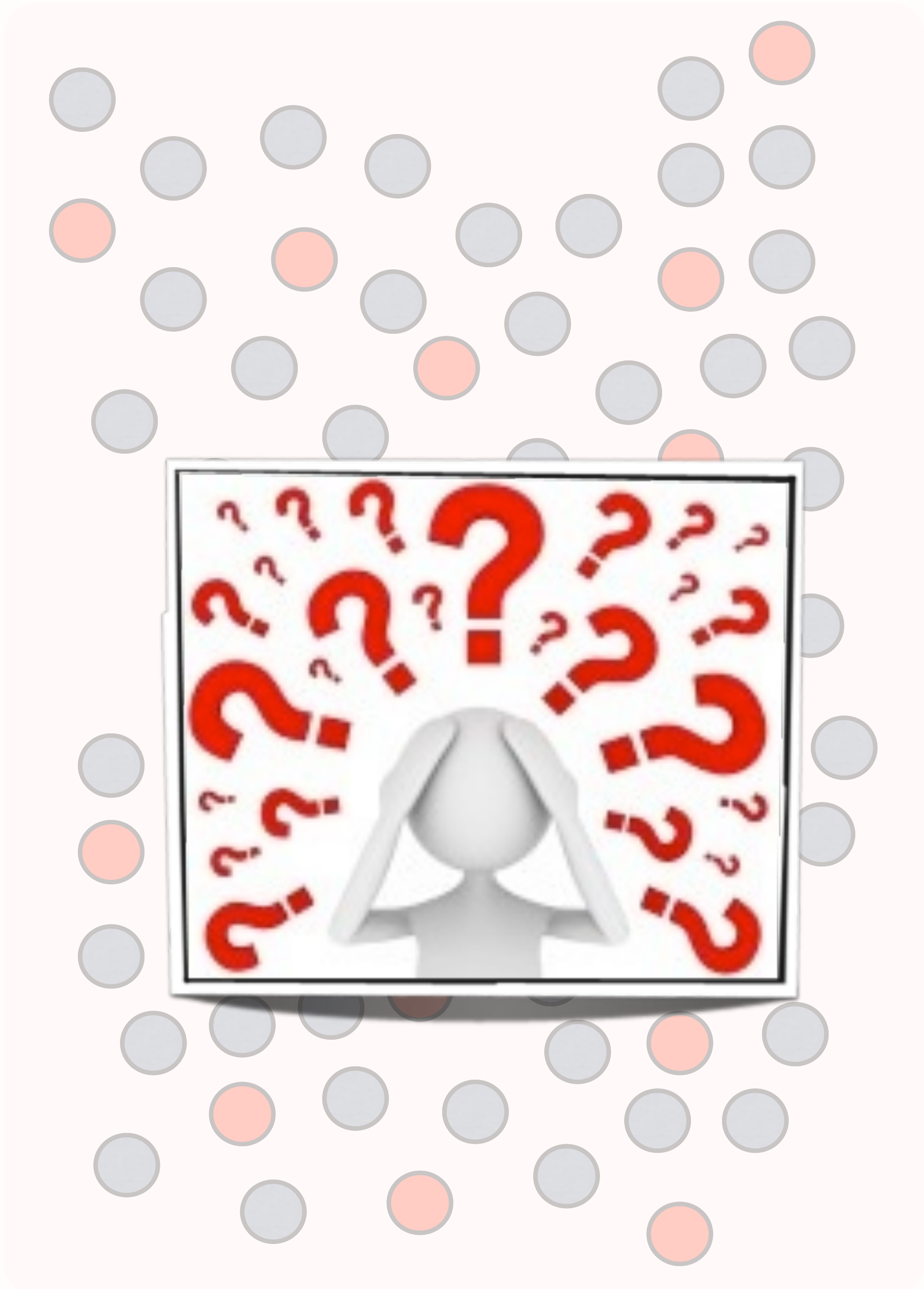


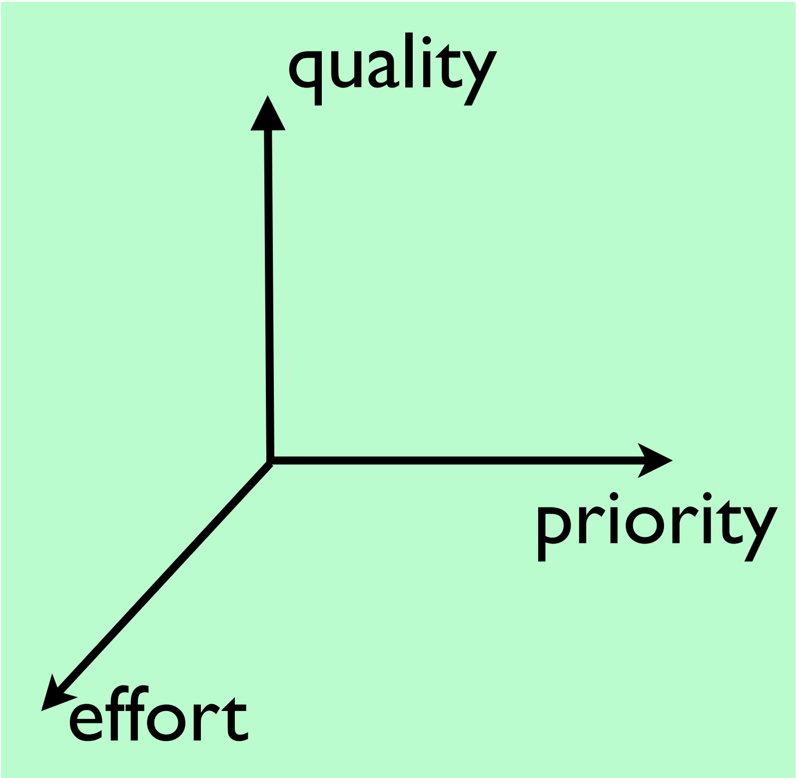
The Problem





The Problem





- order dependency
  [Lee et al. 2010]
- mutual exclusion
  [Liu et al. 2008]
- mutual inclusion
  [Yoshida et al. 2005]

Mathematical Model

$$\mathcal{X}_r = \begin{cases} 0 & \text{if } r \in \mathcal{R} \text{ is not chosen} \\ 1 & \text{if } r \in \mathcal{R} \text{ is chosen} \end{cases} \qquad \mathcal{Y}_r = \begin{cases} 0 & \text{if } r \in \mathcal{R} \text{ is not chosen} \\ k & \text{if } r \in \mathcal{R} \text{ is chosen as the } k^{th} \text{ activity} \end{cases}$$

$$\mathcal{Z}_{ij} = \begin{cases} 0 & \text{if } r_i \perp r_j \\ 1 & \text{if } r_i \nleftrightarrow r_j \\ +2 & \text{if } r_j \nrightarrow r_i \text{ and } r_i \leftrightarrow r_j \\ -2 & \text{if } r_i \nrightarrow r_j \text{ and } r_i \leftrightarrow r_j \\ +3 & \text{if } r_j \nrightarrow r_i\text{, but neither } r_i \leftrightarrow r_j \text{ nor } r_i \nleftrightarrow r_j \\ -3 & \text{if } r_i \nrightarrow r_j\text{, but neither } r_i \leftrightarrow r_j \text{ nor } r_i \nleftrightarrow r_j \end{cases}$$

$$\text{maximize} \quad \sum_{r \in \mathcal{R}} \mathcal{X}_r \rho_r (\overline{Q}_r - E(g_r)) \tag{1}$$

Subject to,

$$\mathcal{X}_r + \mathcal{Y}_r \neq 1, \qquad \forall\, r \in \mathcal{R} \tag{2}$$

$$\mathcal{X}_{r_i} + \mathcal{X}_{r_j} = 2 \Rightarrow \mathcal{Y}_{r_i} \neq \mathcal{Y}_{r_j}, \qquad \forall\, r_i, r_j \in \mathcal{R} \tag{3}$$

$$\mathcal{Z}_{ij} - \mathcal{Z}_{ji} > 0 \Rightarrow \mathcal{Y}_{r_i} < \mathcal{Y}_{r_j}, \qquad \forall\, r_i, r_j \in \mathcal{R} \tag{4}$$

$$\mathcal{Z}_{ij} - \mathcal{Z}_{ji} < 0 \Rightarrow \mathcal{Y}_{r_i} > \mathcal{Y}_{r_j}, \qquad \forall\, r_i, r_j \in \mathcal{R} \tag{5}$$

$$|\mathcal{Z}_{ij}| = 1 \Rightarrow \mathcal{X}_{r_i} + \mathcal{X}_{r_j} \leq 1, \qquad \forall\, r_i, r_j \in \mathcal{R} \tag{6}$$

$$|\mathcal{Z}_{ij}| = 2 \Rightarrow (\mathcal{X}_{r_i} + \mathcal{X}_{r_j})\ modulo\ 2 = 0, \qquad \forall\, r_i, r_j \in \mathcal{R} \tag{7}$$

$$\sum_{r \in \mathcal{R}} \mathcal{X}_r \leq \mathcal{M} \tag{8}$$

Mathematical Model

$$\mathcal{X}_r = \begin{cases} 0 & \text{if } r \in \mathcal{R} \text{ is not chosen} \\ 1 & \text{if } r \in \mathcal{R} \text{ is chosen} \end{cases} \qquad \mathcal{Y}_r = \begin{cases} 0 & \text{if } r \in \mathcal{R} \text{ is not chosen} \\ k & \text{if } r \in \mathcal{R} \text{ is chosen as the } k^{th} \text{ activity} \end{cases}$$

$$\mathcal{Z}_{ij} = \begin{cases} 0 & \text{if } r_i \perp r_j \\ 1 & \text{if } r_i \nleftrightarrow r_j \\ +2 & \text{if } r_j \nrightarrow r_i \text{ and } r_i \leftrightarrow r_j \\ -2 & \text{if } r_i \nrightarrow r_j \text{ and } r_i \leftrightarrow r_j \\ +3 & \text{if } r_j \nrightarrow r_i\text{, but neither } r_i \leftrightarrow r_j \text{ nor } r_i \nleftrightarrow r_j \\ -3 & \text{if } r_i \nrightarrow r_j\text{, but neither } r_i \leftrightarrow r_j \text{ nor } r_i \nleftrightarrow r_j \end{cases}$$

Objective function

$$\text{maximize} \quad \sum_{r \in \mathcal{R}} \mathcal{X}_r \rho_r (\overline{Q}_r - E(g_r)) \qquad (1)$$

Subject to,

$$\mathcal{X}_r + \mathcal{Y}_r \neq 1, \qquad \forall\, r \in \mathcal{R} \quad (2)$$

$$\mathcal{X}_{r_i} + \mathcal{X}_{r_j} = 2 \Rightarrow \mathcal{Y}_{r_i} \neq \mathcal{Y}_{r_j}, \qquad \forall\, r_i, r_j \in \mathcal{R} \quad (3)$$

$$\mathcal{Z}_{ij} - \mathcal{Z}_{ji} > 0 \Rightarrow \mathcal{Y}_{r_i} < \mathcal{Y}_{r_j}, \qquad \forall\, r_i, r_j \in \mathcal{R} \quad (4)$$

$$\mathcal{Z}_{ij} - \mathcal{Z}_{ji} < 0 \Rightarrow \mathcal{Y}_{r_i} > \mathcal{Y}_{r_j}, \qquad \forall\, r_i, r_j \in \mathcal{R} \quad (5)$$

$$|\mathcal{Z}_{ij}| = 1 \Rightarrow \mathcal{X}_{r_i} + \mathcal{X}_{r_j} \leq 1, \qquad \forall\, r_i, r_j \in \mathcal{R} \quad (6)$$

$$|\mathcal{Z}_{ij}| = 2 \Rightarrow (\mathcal{X}_{r_i} + \mathcal{X}_{r_j})\ modulo\ 2 = 0, \qquad \forall\, r_i, r_j \in \mathcal{R} \quad (7)$$

$$\sum_{r \in \mathcal{R}} \mathcal{X}_r \leq \mathcal{M} \qquad (8)$$

Mathematical Model

$$\mathcal{X}_r = \begin{cases} 0 & \text{if } r \in \mathcal{R} \text{ is not chosen} \\ 1 & \text{if } r \in \mathcal{R} \text{ is chosen} \end{cases} \qquad \mathcal{Y}_r = \begin{cases} 0 & \text{if } r \in \mathcal{R} \text{ is not chosen} \\ k & \text{if } r \in \mathcal{R} \text{ is chosen as the } k^{th} \text{ activity} \end{cases}$$

$$\mathcal{Z}_{ij} = \begin{cases} 0 & \text{if } r_i \perp r_j \\ 1 & \text{if } r_i \not\leftrightarrow r_j \\ +2 & \text{if } r_j \not\rightarrow r_i \text{ and } r_i \leftrightarrow r_j \\ -2 & \text{if } r_i \not\rightarrow r_j \text{ and } r_i \leftrightarrow r_j \\ +3 & \text{if } r_j \not\rightarrow r_i\text{, but neither } r_i \leftrightarrow r_j \text{ nor } r_i \not\leftrightarrow r_j \\ -3 & \text{if } r_i \not\rightarrow r_j\text{, but neither } r_i \leftrightarrow r_j \text{ nor } r_i \not\leftrightarrow r_j \end{cases}$$

Objective function

$$\text{maximize} \quad \sum_{r \in \mathcal{R}} \mathcal{X}_r \rho_r (\overline{Q}_r - E(g_r)) \tag{1}$$

Constraints

Subject to,

$$\mathcal{X}_r + \mathcal{Y}_r \neq 1, \qquad \forall\ r \in \mathcal{R} \tag{2}$$

$$\mathcal{X}_{r_i} + \mathcal{X}_{r_j} = 2 \Rightarrow \mathcal{Y}_{r_i} \neq \mathcal{Y}_{r_j}, \qquad \forall\ r_i, r_j \in \mathcal{R} \tag{3}$$

$$\mathcal{Z}_{ij} - \mathcal{Z}_{ji} > 0 \Rightarrow \mathcal{Y}_{r_i} < \mathcal{Y}_{r_j}, \qquad \forall\ r_i, r_j \in \mathcal{R} \tag{4}$$

$$\mathcal{Z}_{ij} - \mathcal{Z}_{ji} < 0 \Rightarrow \mathcal{Y}_{r_i} > \mathcal{Y}_{r_j}, \qquad \forall\ r_i, r_j \in \mathcal{R} \tag{5}$$

$$|\mathcal{Z}_{ij}| = 1 \Rightarrow \mathcal{X}_{r_i} + \mathcal{X}_{r_j} \leq 1, \qquad \forall\ r_i, r_j \in \mathcal{R} \tag{6}$$

$$|\mathcal{Z}_{ij}| = 2 \Rightarrow (\mathcal{X}_{r_i} + \mathcal{X}_{r_j})\ modulo\ 2 = 0, \qquad \forall\ r_i, r_j \in \mathcal{R} \tag{7}$$

$$\sum_{r \in \mathcal{R}} \mathcal{X}_r \leq \mathcal{M} \tag{8}$$

Implementation

- Constraint Programming
- OPL Implementation
- IBM ILOG CPLEX Optimization Studio 12.2

Effort Estimation

- Refactoring Effort Model
  - Understanding the Context
    - method delegation
    - inheritance hierarchy
  - Code Modification Effort
    - token modification
    - code relocation
  - Navigation Effort
    - Dispersion of source files

Effect Estimation

Attribute		Formula
Reusability	=	$-0.25 \times \text{DCC} + 0.25 \times \text{CAM} + 0.5 \times \text{CIS} + 0.5 \times \text{DSC}$
Flexibility	=	$0.25 \times \text{DAM} - 0.25 \times \text{DCC} + 0.5 \times \text{MOA} + 0.5 \times \text{NOP}$
Understandability	=	$-0.33 \times \text{ANA} + 0.33 \times \text{DAM} - 0.33 \times \text{DCC} + 0.33 \times \text{CAM} - 0.33 \times \text{NOP} - 0.33 \times \text{NOM} - 0.33 \times \text{DSC}$
Functionality	=	$0.12 \times \text{CAM} + 0.22 \times \text{NOP} + 0.22 \times \text{CIS} + 0.22 \times \text{DSC} + 0.22 \times \text{NOH}$
Extendability	=	$0.5 \times \text{ANA} - 0.5 \times \text{DCC} + 0.5 \times \text{MFA} + 0.5 \times \text{NOP}$
Effectiveness	=	$0.2 \times \text{ANA} + 0.2 \times \text{DAM} + 0.2 \times \text{MOA} + 0.2 \times \text{MFA} + 0.2 \times \text{NOP}$

QMOOD

[Bansiya and Davis, 2002]

Design Property	Metric	Description
Design size	DSC	Design size in classes
Complexity	NOM	Number of methods
Coupling	DCC	Direct class coupling
Polymorphism	NOP	Number of polymorphic methods
Hierarchies	NOH	Number of hierarchies
Cohesion	CAM	Cohesion among methods in class
Abstraction	ANA	Average number of ancestors
Encapsulation	DAM	Data access metric
Composition	MOA	Measure of aggregation
Inheritance	MFA	Measure of functional abstraction
Messaging	CIS	Class interface size

Empirical Evaluation

Subject Systems	Clone Groups	Clone Fragments	Total Refactorings
Mutation Framework	21	62	72
LIME	20	55	67
gCad	28	91	93
VisCad	57	136	166

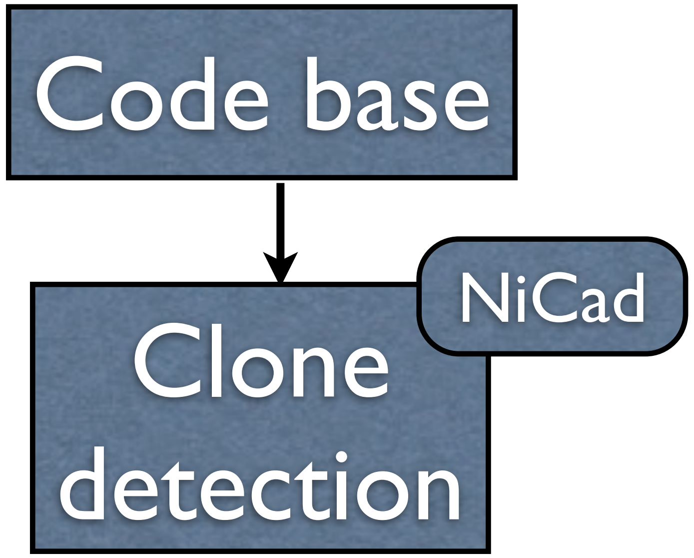
- Validation with manual approach
- Comparison with variants of greedy approach

Evaluation Procedure

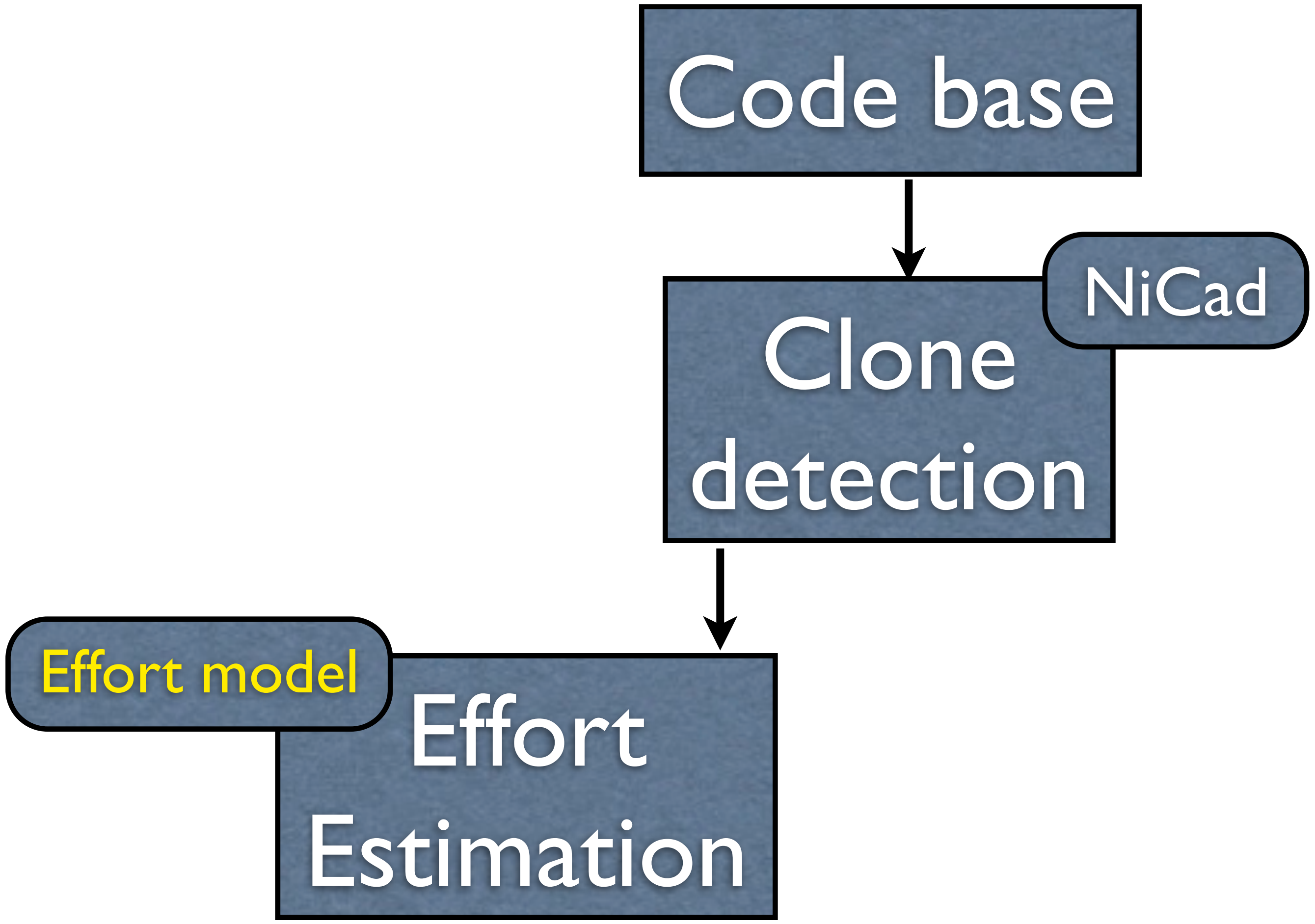
Evaluation Procedure

Code base

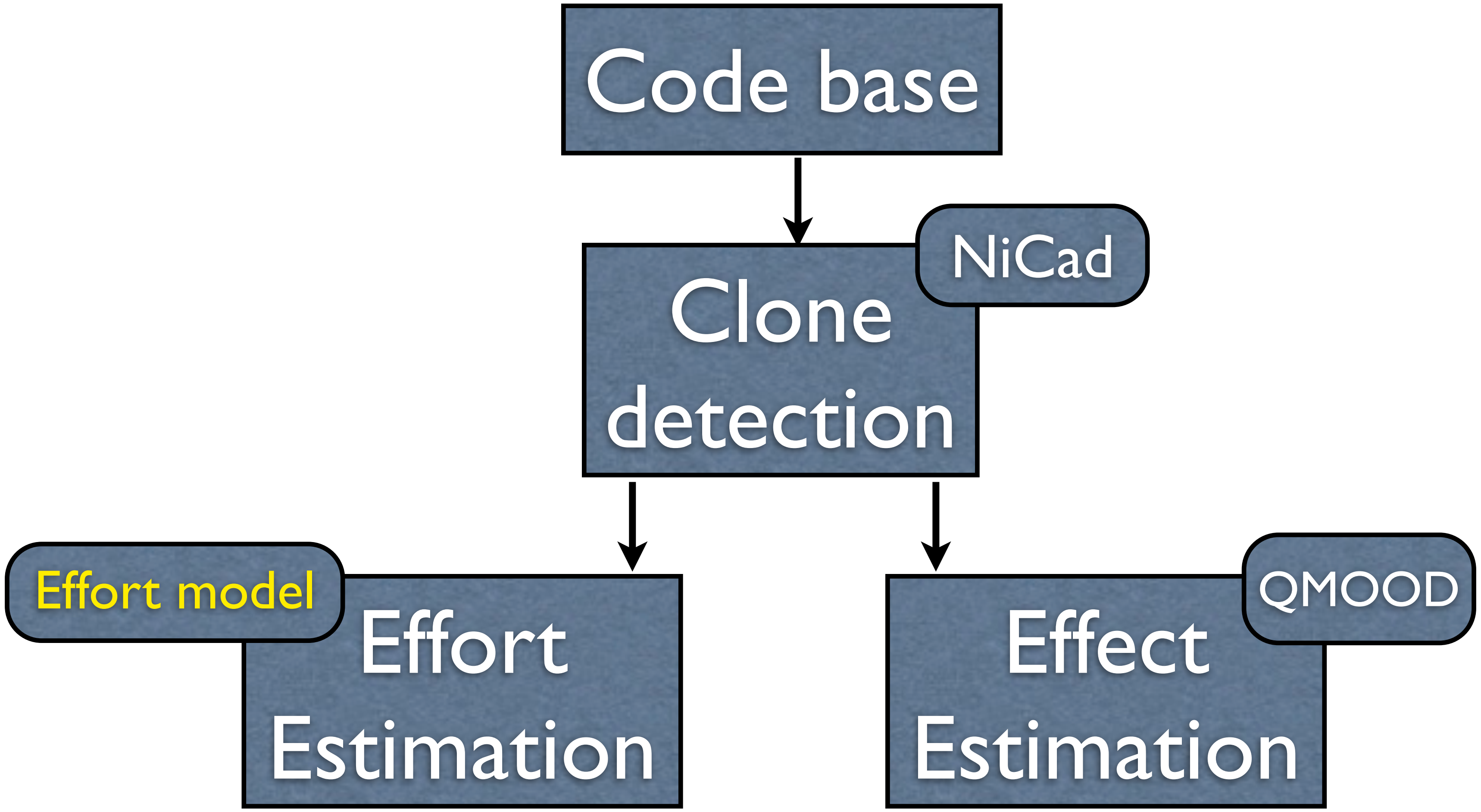
Evaluation Procedure



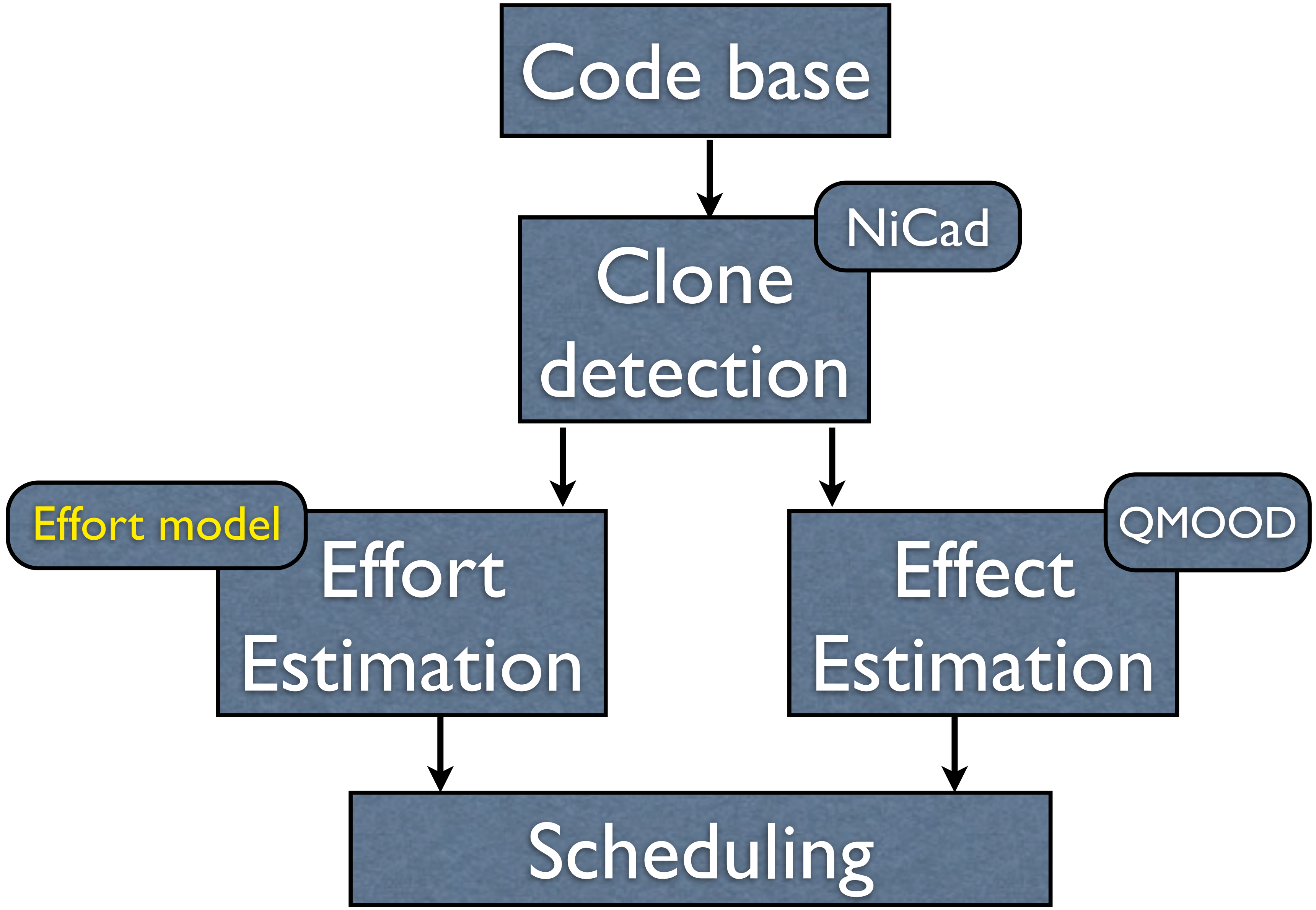
Evaluation Procedure



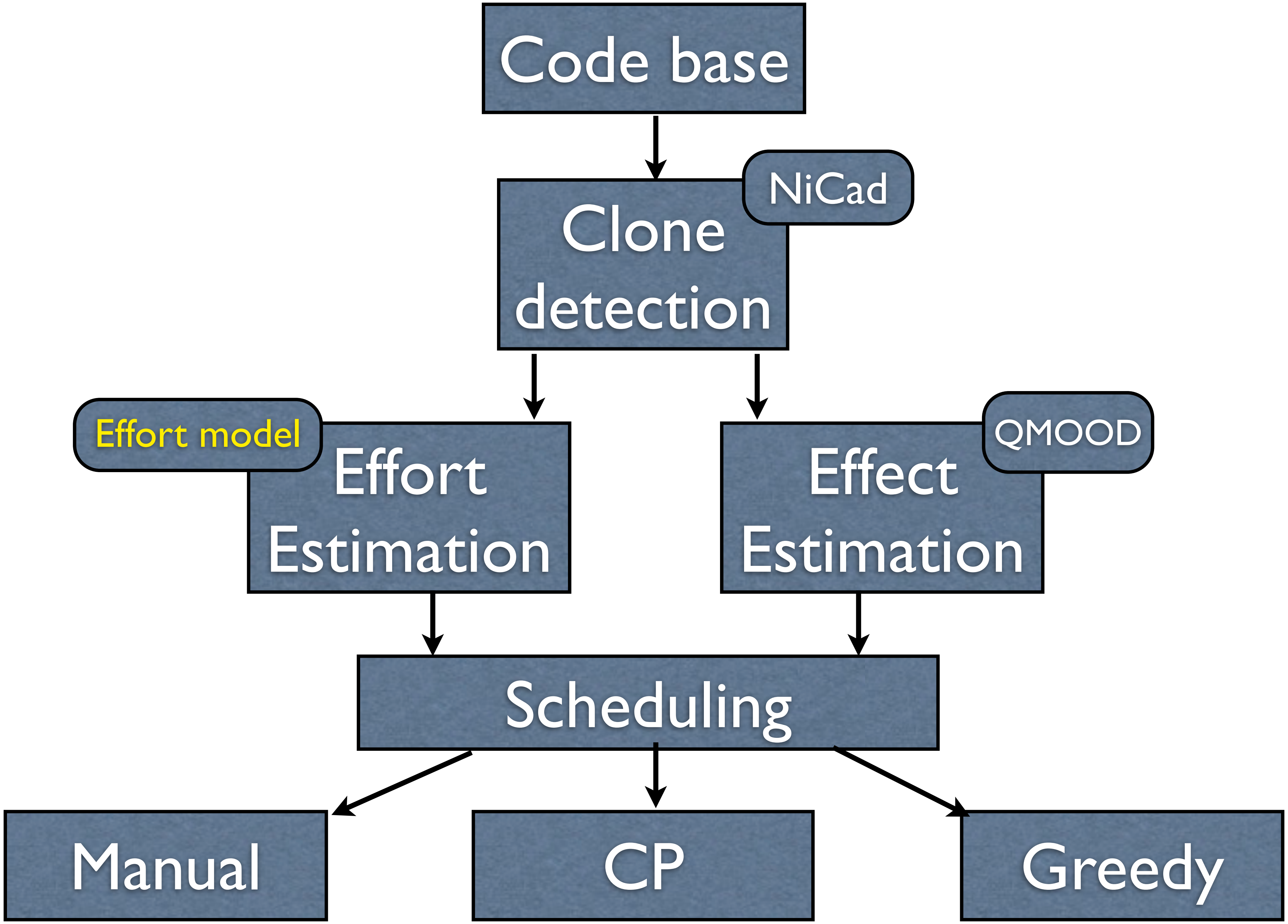
Evaluation Procedure



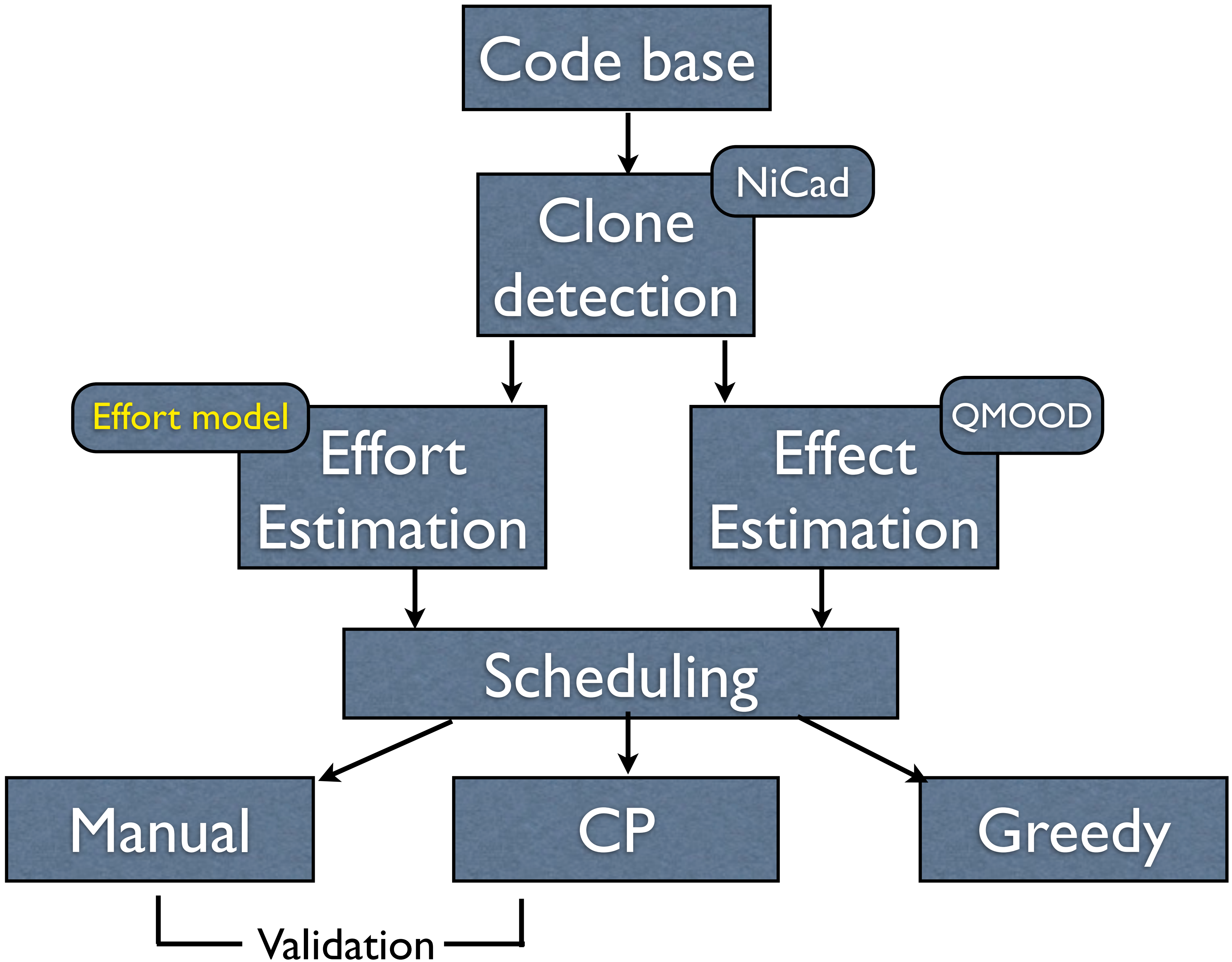
Evaluation Procedure



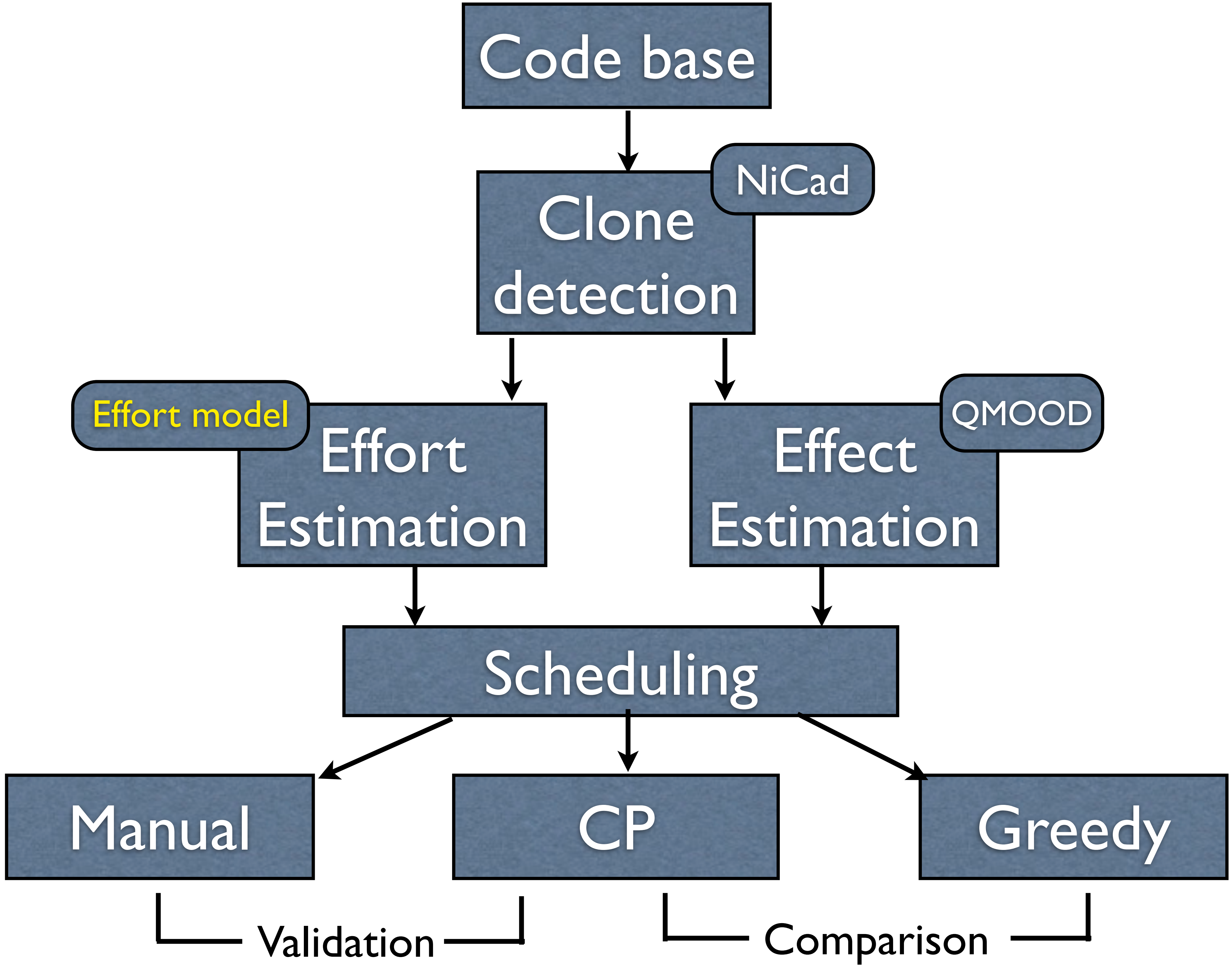
Evaluation Procedure



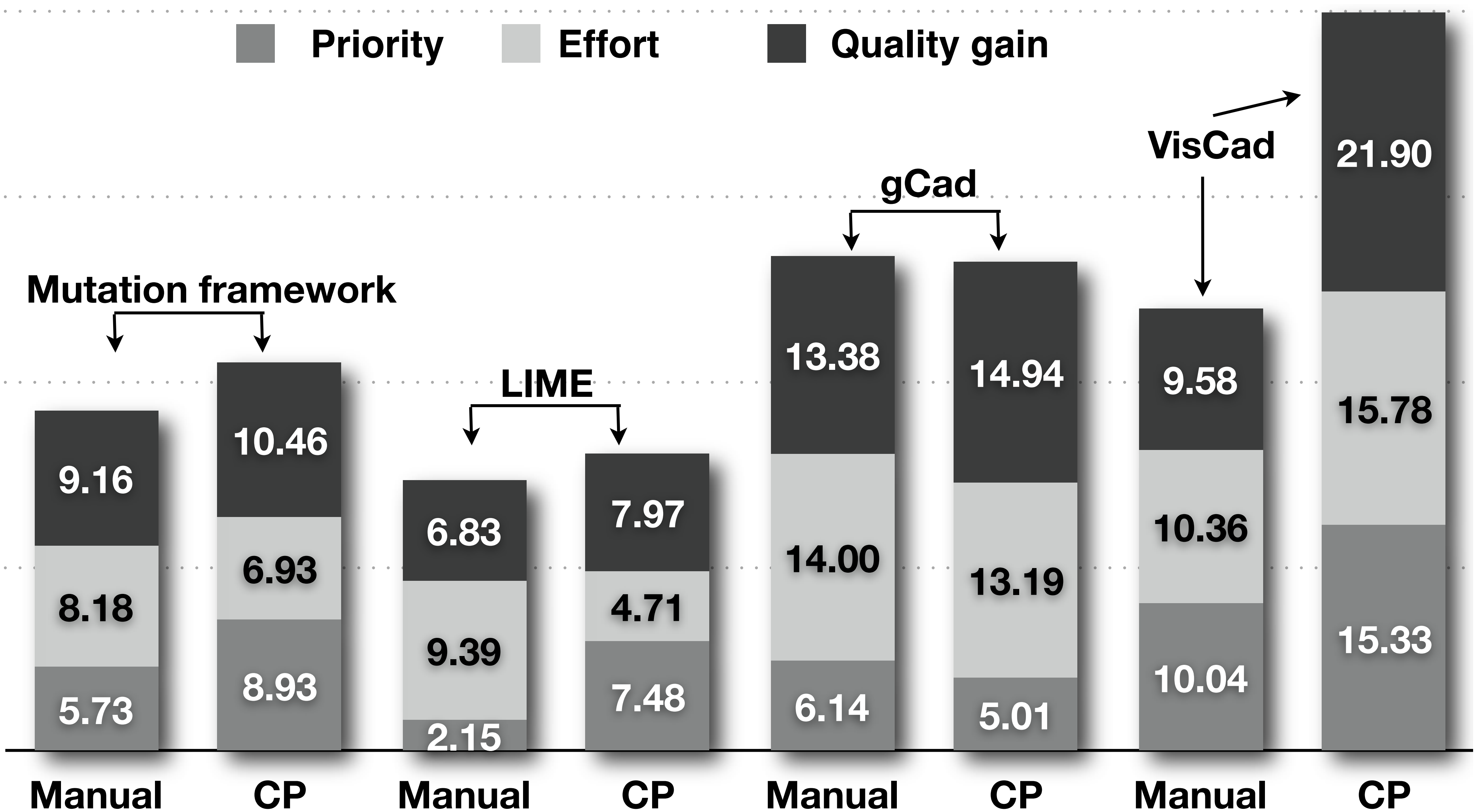
Evaluation Procedure



Evaluation Procedure



Results


Priority
Effort
Quality gain
Mutation framework
LIME
gCad
VisCad
9.16
8.18
5.73
10.46
6.93
8.93
6.83
9.39
2.15
7.97
4.71
7.48
13.38
14.00
6.14
14.94
13.19
5.01
9.58
10.36
10.04
21.90
15.78
15.33
Manual
CP
Manual
CP
Manual
CP
Manual
CP

Results

COMPARISON OF CP AND GREEDY SCHEDULING APPROACHES

Subject systems	**Scheduling approaches**	**Values at dimensions**			**Refac. chosen**
		Prior.	**Effort**	**Quality**	
Mutation Frame-work	$Greedy^p$	20.06	21.94	18.53	40
	$Greedy^e$	9.63	6.06	10.04	20
	$Greedy^q$	18.16	21.82	19.64	42
	CP	9.34	7.86	11.48	20
LIME	$Greedy^p$	22.42	21.12	19.93	47
	$Greedy^e$	13.00	8.28	13.61	33
	$Greedy^q$	16.29	23.49	26.07	51
	CP	11.04	12.32	16.12	33
gCad	$Greedy^p$	19.65	21.62	20.00	41
	$Greedy^e$	9.61	9.53	11.57	28
	$Greedy^q$	12.05	23.48	25.98	44
	CP	6.69	15.19	17.99	28
VisCad	$Greedy^p$	36.14	32.57	25.71	66
	$Greedy^e$	16.12	18.63	13.20	40
	$Greedy^q$	29.02	33.81	34.32	72
	CP	15.33	15.78	21.90	40

Here, $Greedy^p$ = approach greedy towards priority satisfaction
$Greedy^e$ = approach greedy towards effort minimization
$Greedy^q$ = approach greedy towards quality gain

Related Work

Refactoring scheduler	Scheduling approach	Refactoring effort	Quality gain	Sequential dependency	Mutual exclusion	Mutual inclusion	Priorities satisfaction
Bouktif et. al. [5]	GA	√	√				
Lee et. al. [20]	OmeGA		√	√	√		
Liu et. al. [19]	Heuristic		√	√	√		
Our Scheduler	CP	√	√	√	√	√	√

[5] S. Bouktif, G. Antoniol, M. Neteler, and E. Merlo. A Novel Approach to Optimize Clone Refactoring Activity. In *GECCO*, July 8 –12, 2006.

[19] H. Liu, G. Li, Z. Ma, and W. Shao. Conflict-aware schedule of software refactorings. *IET Softw.*, 2(5): 446–460, 2008.

[20] S. Lee, G. Bae, H. S. Chae, and D. Bae, and Yong Rae Kwon. Automated scheduling for clone-based refactoring using a competent GA. *Softw. Pract. Exper.*, Wiley Online Library, 2010.

Contributions

- Novel Effort Model
- Novel Approach (CP)
- Wide Range of Constraints
- Captures Risks