software traceability

Managing Traceability and Traceability Recovery

PART I
Understanding the Problem

Managing Traceability and Traceability Recovery

PART I
Understanding the Problem

PART II
Keeping Traceability Links

Managing Traceability and Traceability Recovery

PART I
Understanding the Problem

PART II
Keeping Traceability Links

PART III
Traceability Dimensions

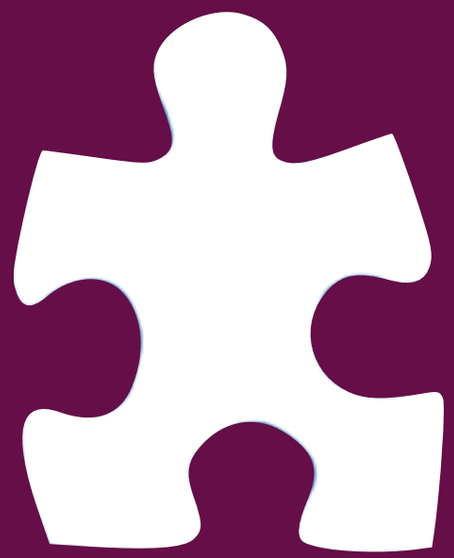
Managing Traceability and Traceability Recovery

PART I
Understanding the Problem

PART II
Keeping Traceability Links

PART III
Traceability Dimensions

PART IV
Traceability Recovery



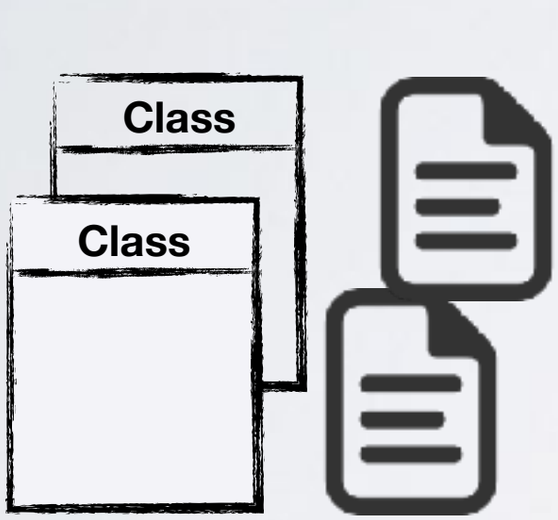
PART I
Understanding the Problem

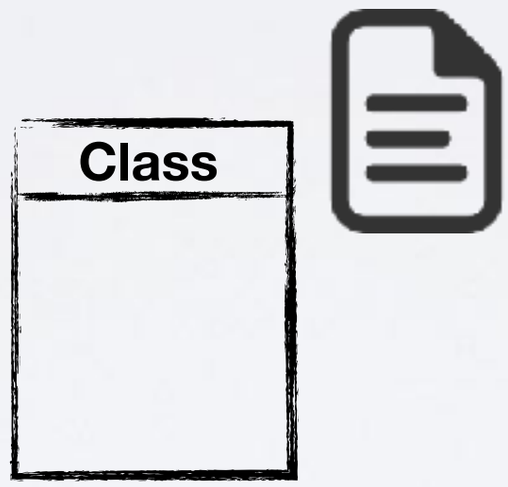
What does it make a software project successful?

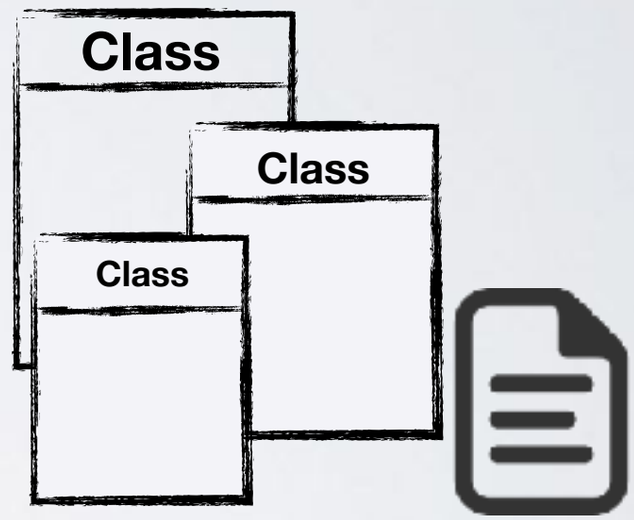
What does it make a software project successful?

All the expected requirements should be implemented

Requirement **Requirement** ... **Requirement**





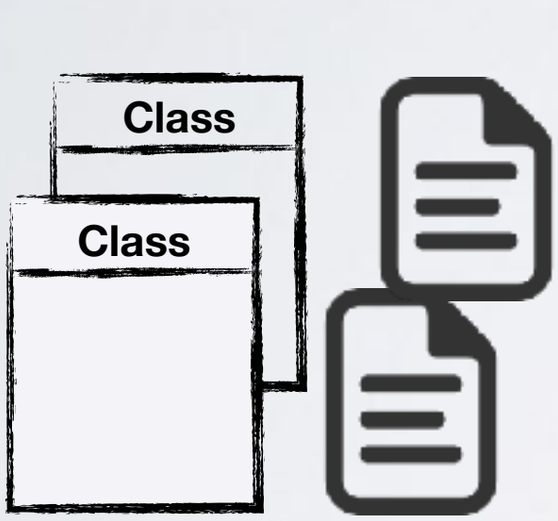


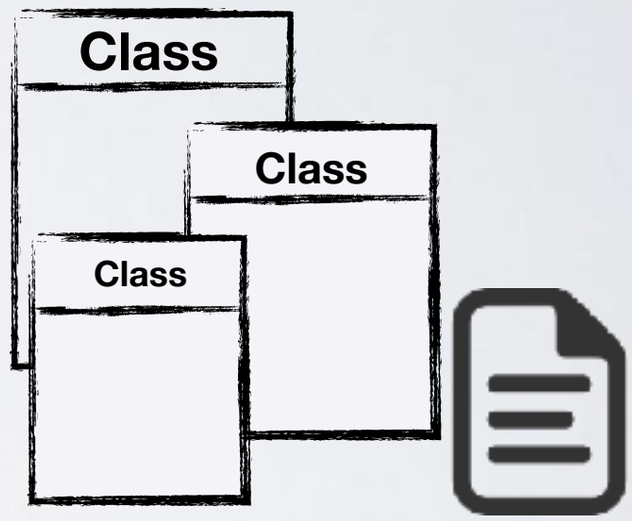
Need to **control** how requirements evolve

What does it make a software project successful?

All the expected requirements should be implemented

Requirement **Requirement** ... **Requirement**

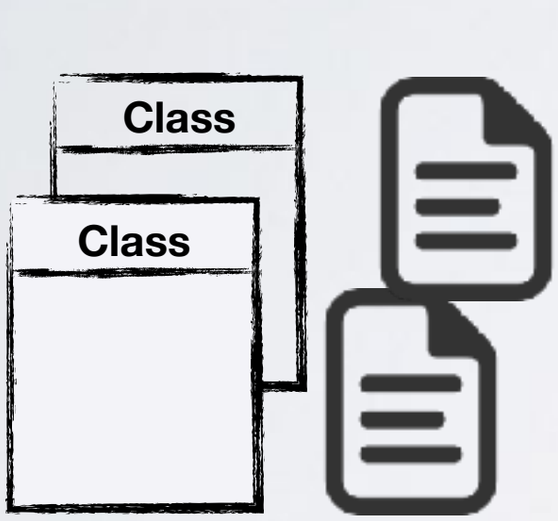


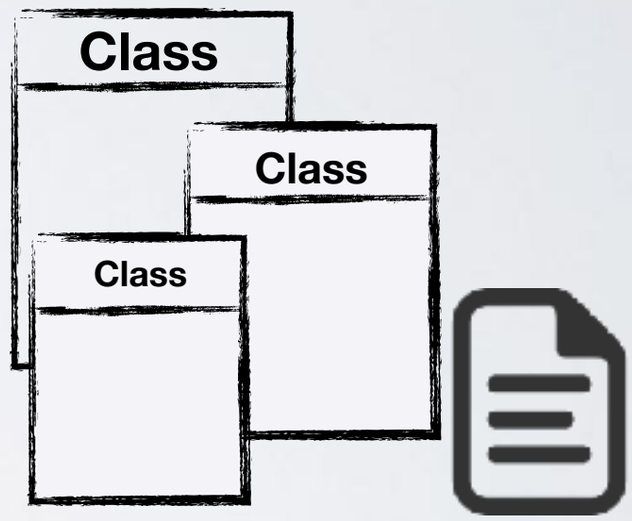


What does it make a software project successful?

All the expected requirements should be implemented

Requirement **Requirement** ... **Requirement**





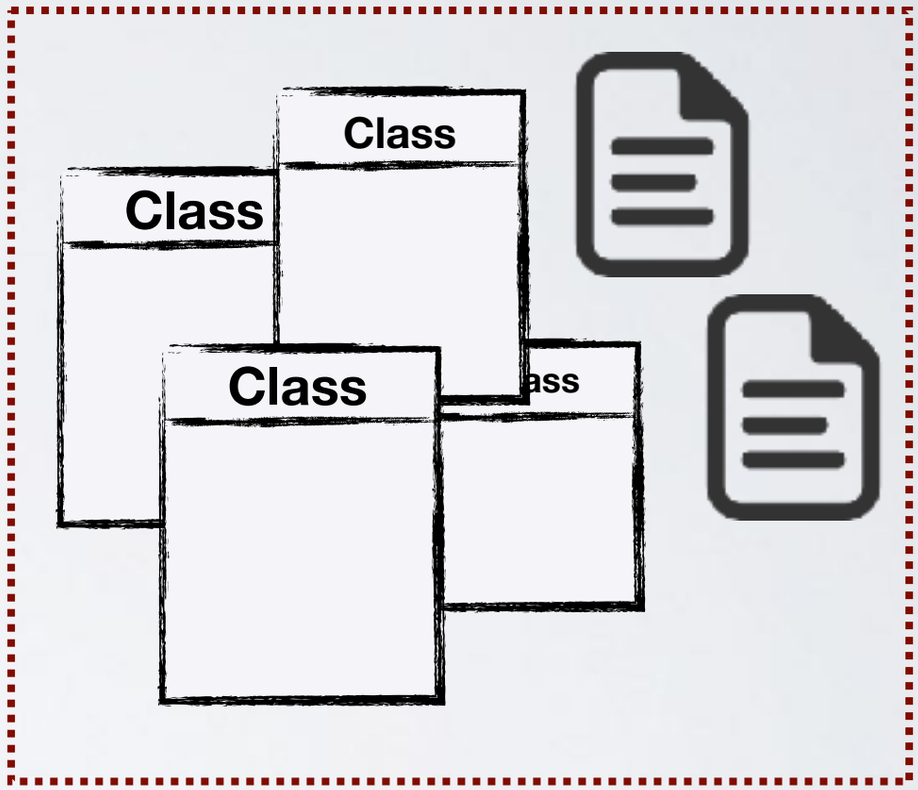
Management of the **risks** associated with missing requirements

During this course you will be requested to change your project several times: what would be the first operation to apply a change request?

During this course you will be requested to change your project several times: what would be the first operation to apply a change request?

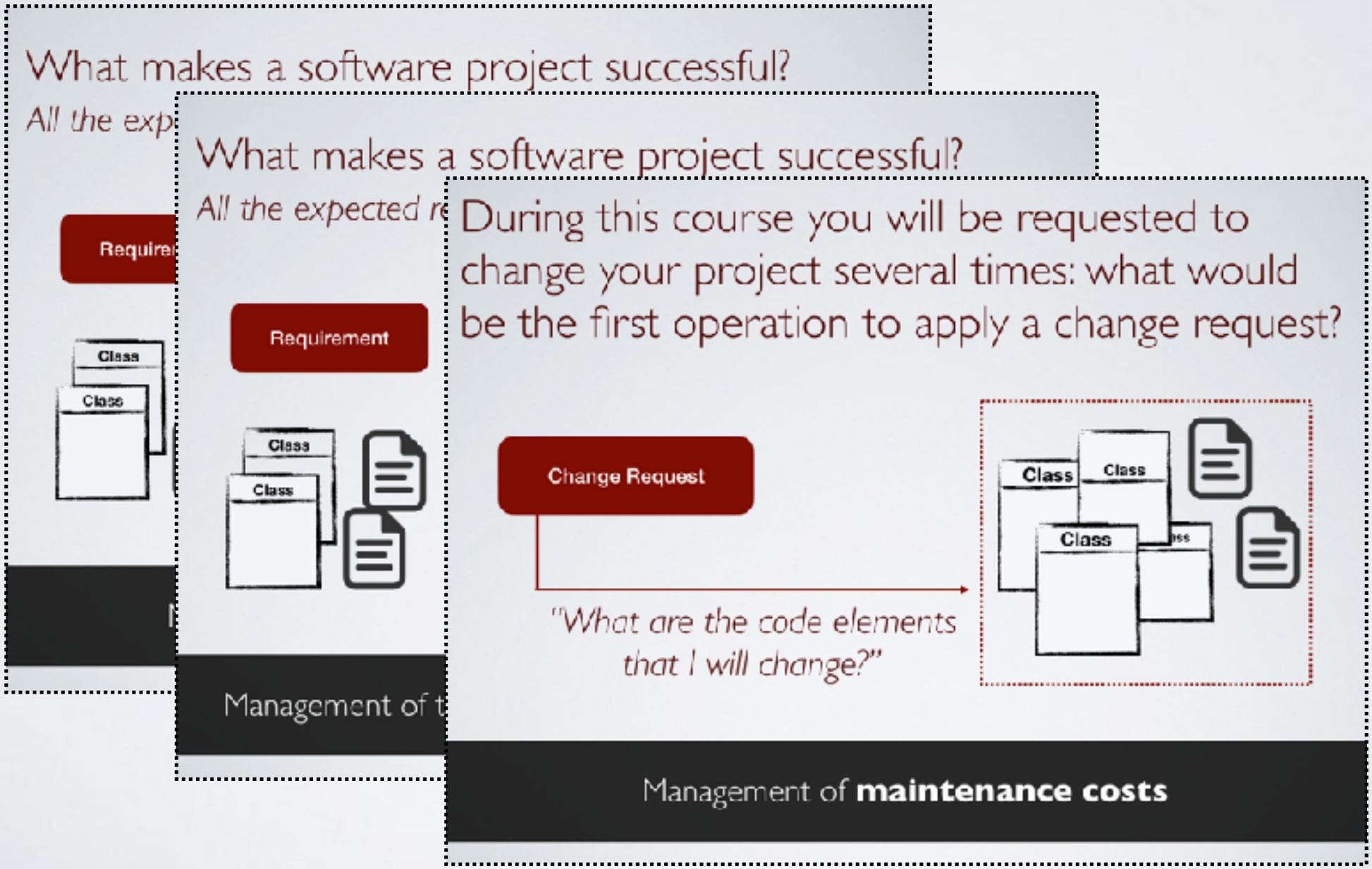
Change Request

"What are the code elements that I will change?"



Management of **maintenance costs**

Traceability can support all these activities!



Traceability can support all these activities!

Traceability shows forward and backward relationships linking requirements with design, implementation, test, and maintenance

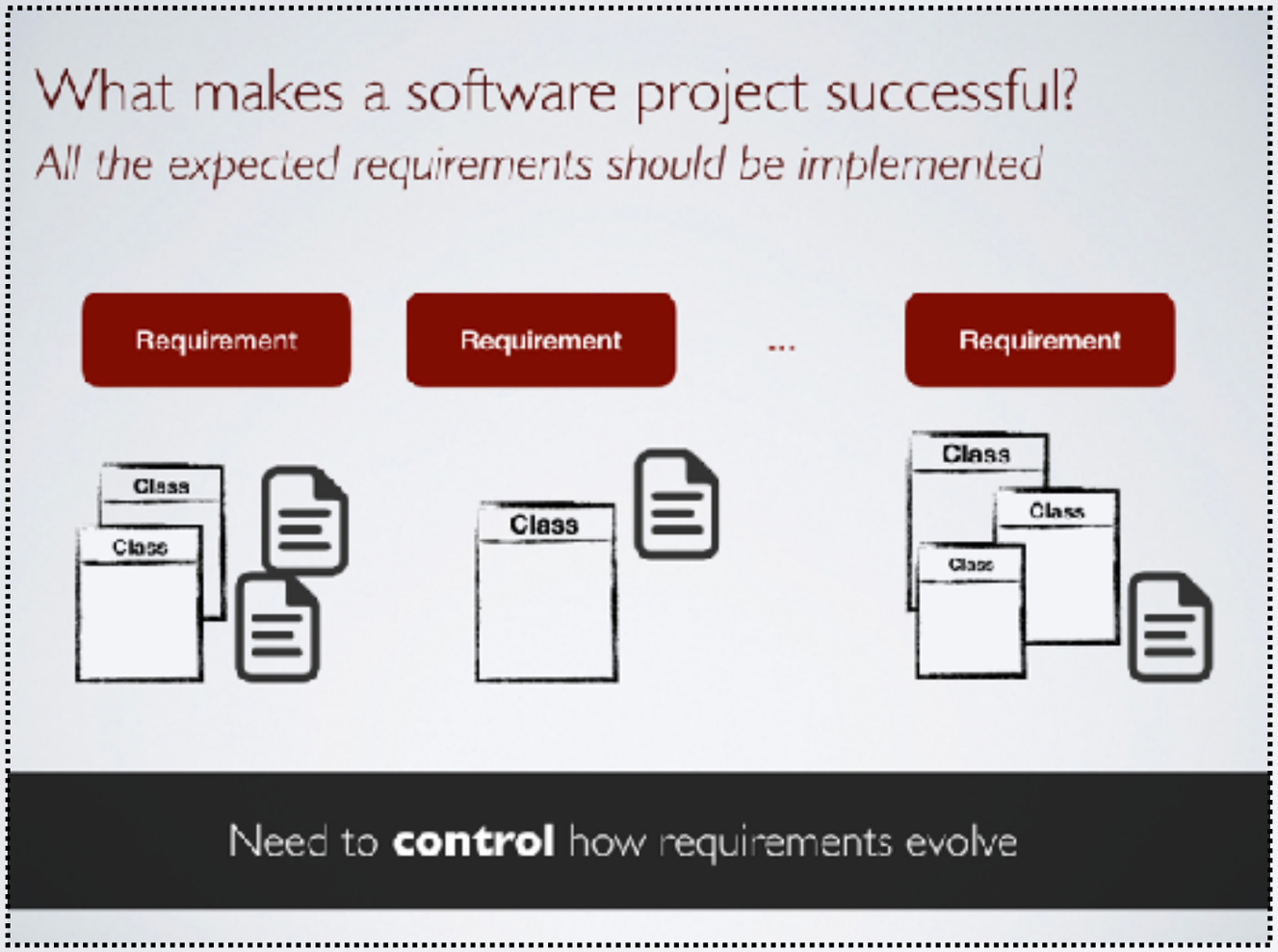
Traceability can support all these activities!

A forward traceability link reveals the existence of a relationship between a higher-level artifact (e.g., a requirement) and a lower-level one (e.g., a source code class)

Traceability can support all these activities!

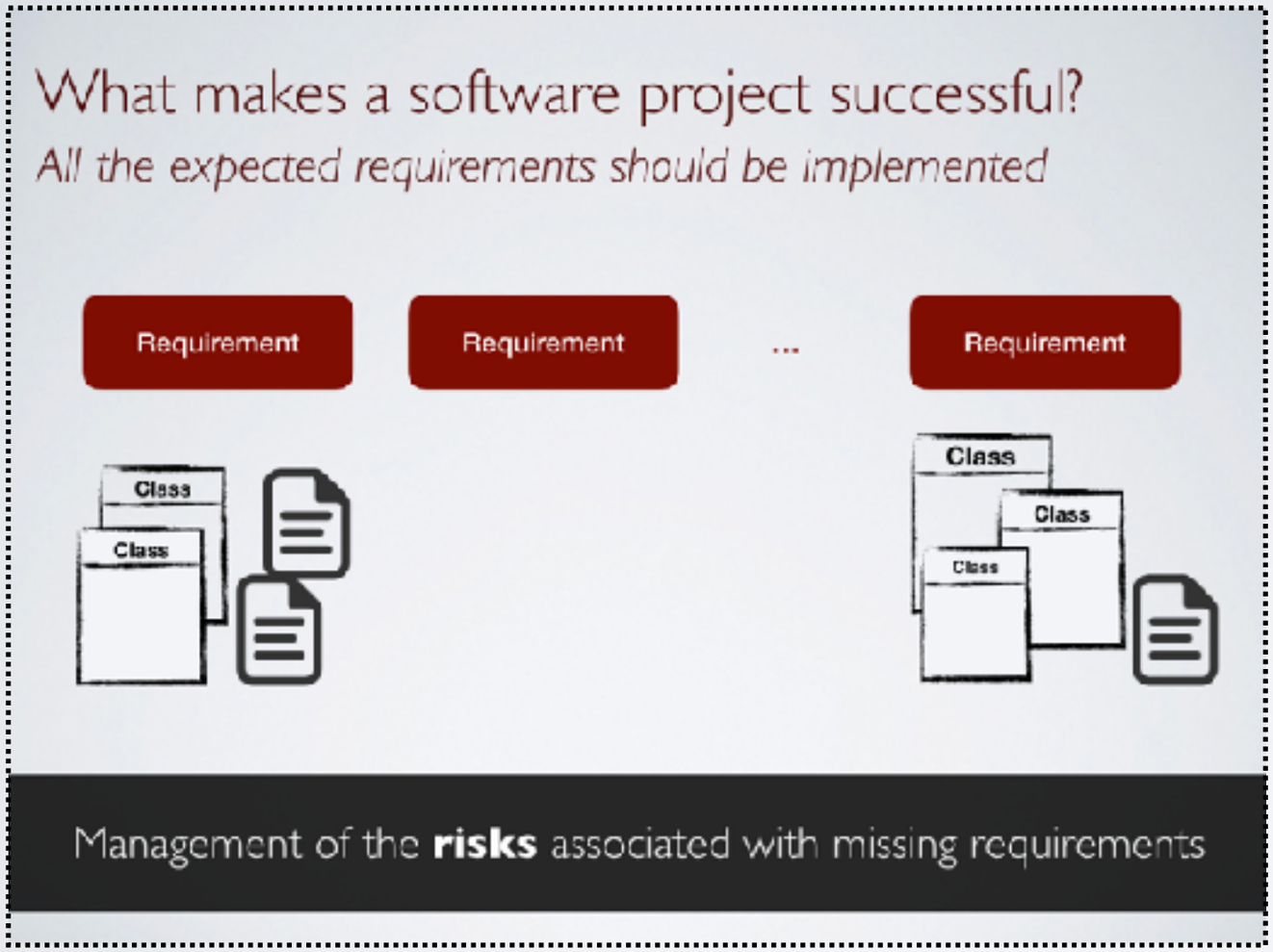
A backward traceability link reveals the existence of a relationship between a lower-level artifact (e.g., a source code class) and a higher-level one (e.g., a requirement)

Traceability can support all these activities!



Forward traceability links can be used to control the evolution of requirements

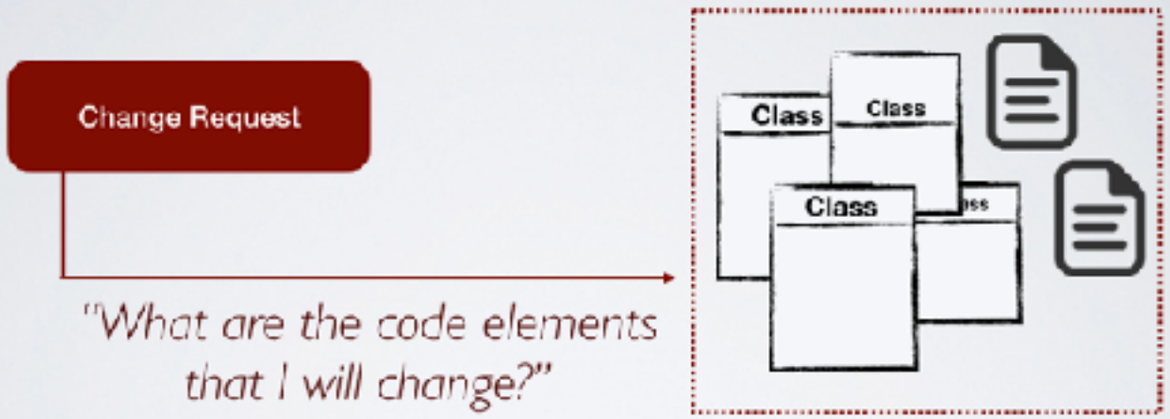
Traceability can support all these activities!



Missing traceability links between a requirement and other artifacts can reveal missing requirements

Traceability can support all these activities!

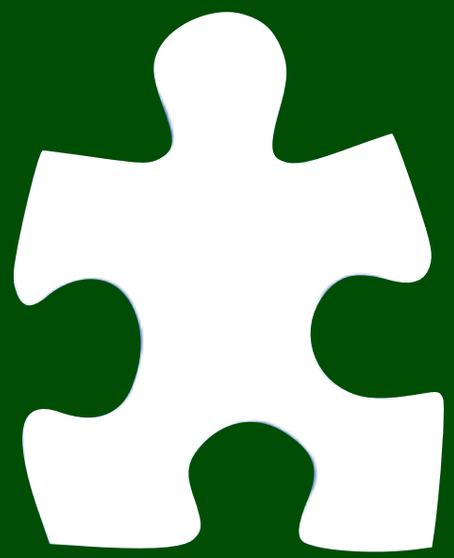
During this course you will be requested to change your project several times: what would be the first operation to apply a change request?



Management of **maintenance costs**

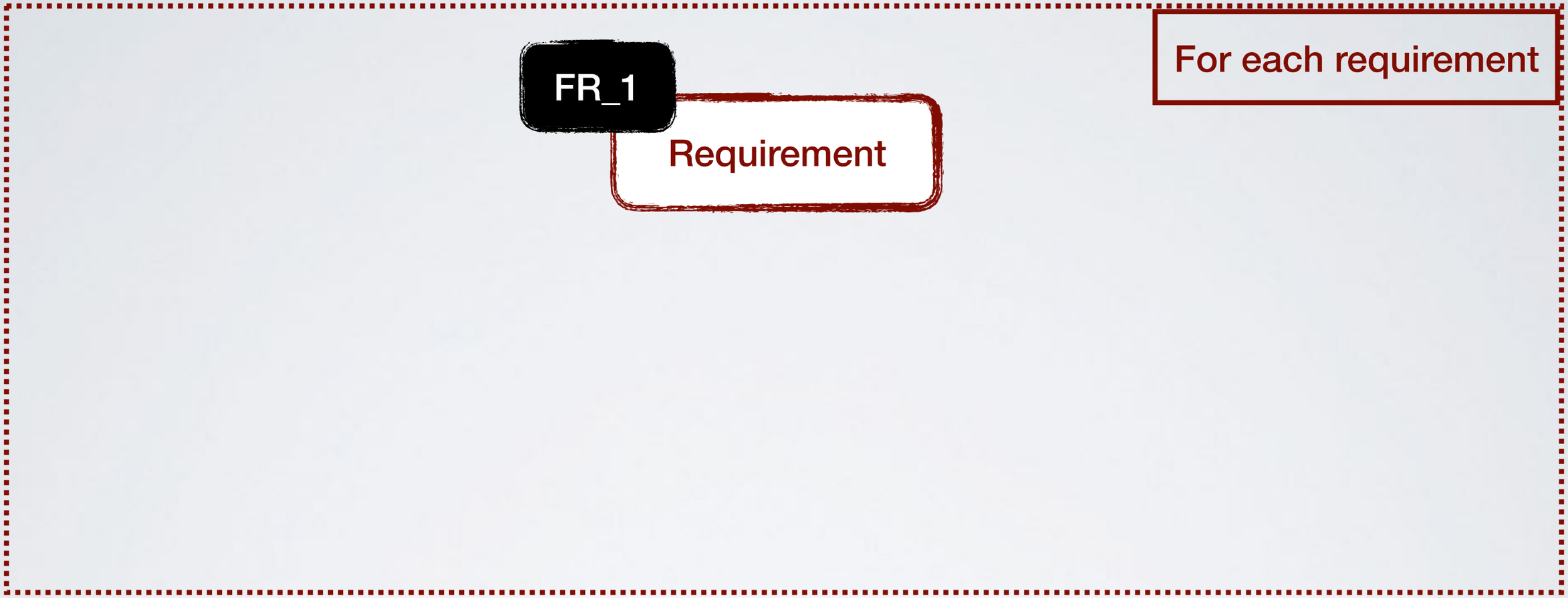
A manager can evaluate the cost of a change request by using traceability links

Keeping and maintaining traceability links during the entire lifecycle

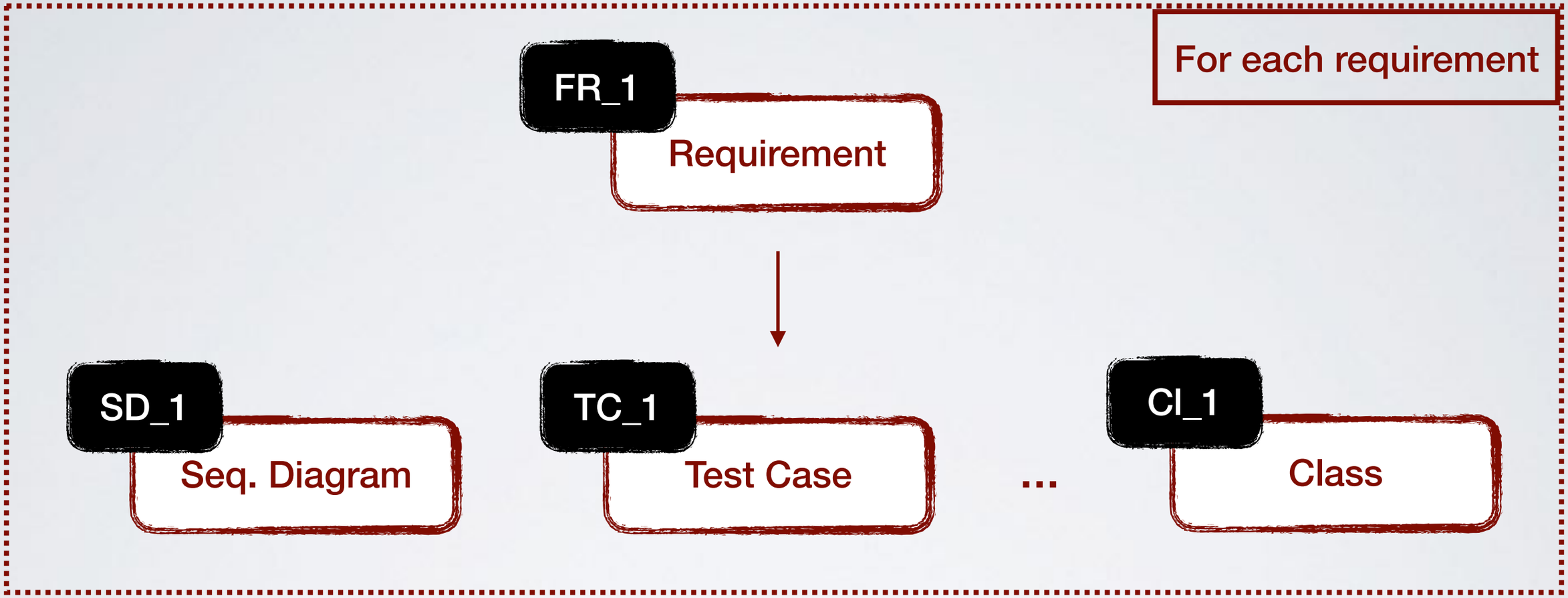


PART II
Keeping Traceability Links

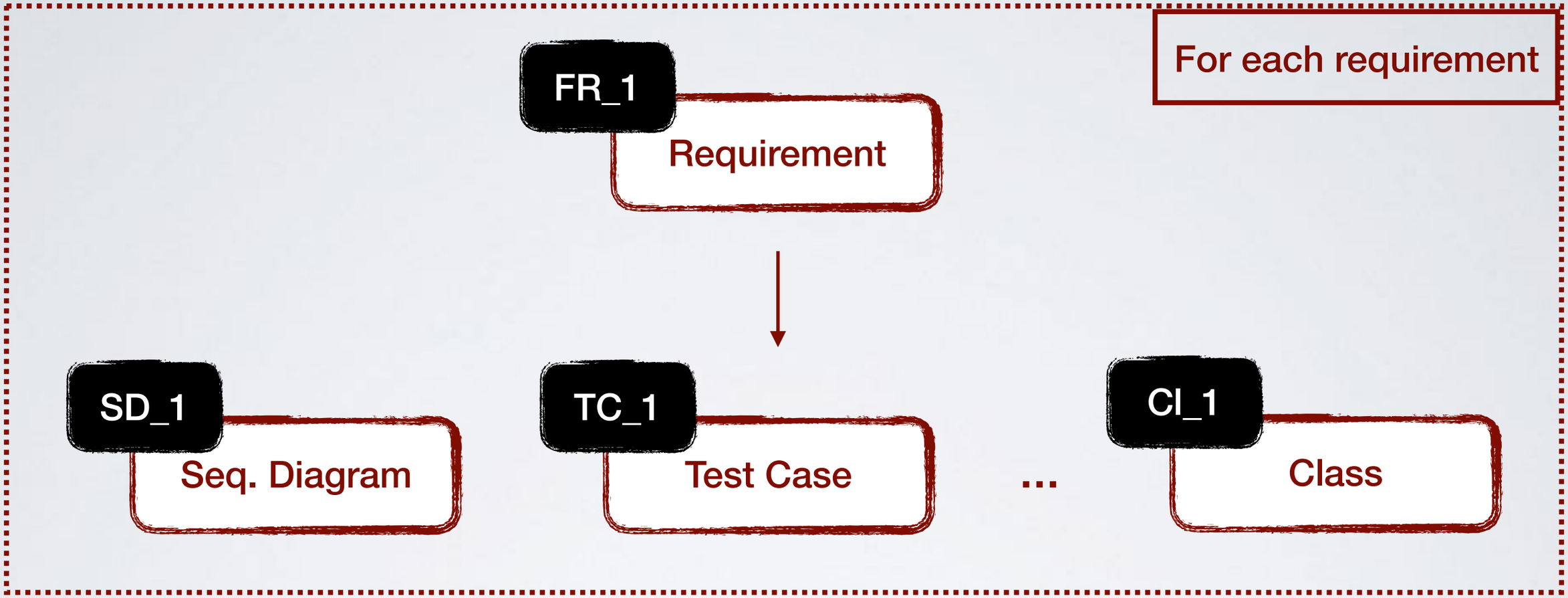
Keeping Traceability Links: The Process



Keeping Traceability Links: The Process

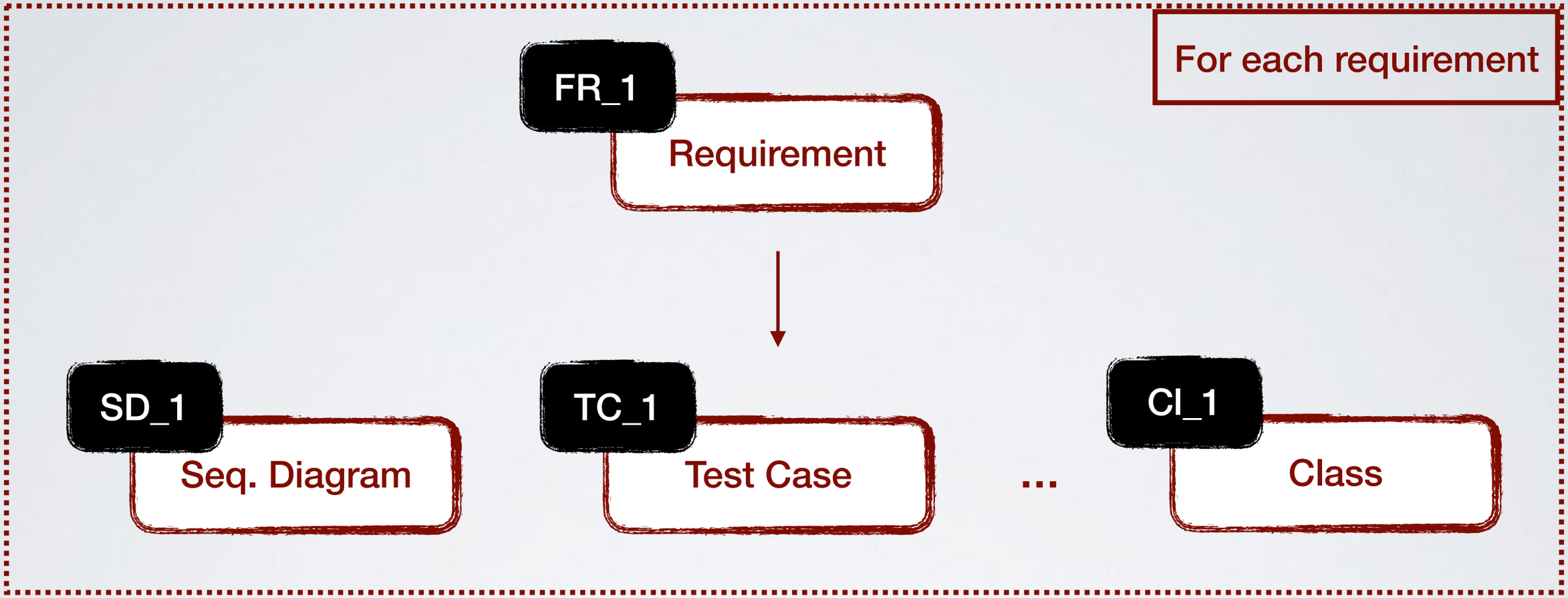


Keeping Traceability Links: The Process



Process done manually, even if it can be managed using tools

Keeping Traceability Links: The Process



Process done manually, even if it can be managed using tools

Relationships between related requirements cannot be easily managed

Keeping Traceability Links: The Process

Requirement ID	REQ-1 UC 1.1	REQ-1 UC 1.2	REQ-1 SD 1.1	REQ-1 SD 1.2	REQ-1 Class 1.1	REQ-1 Class 1.2	REQ-1 TC 1.1	REQ-1 TC 1.2
REQ-1 UC 1.1								
REQ-1 UC 1.2								
REQ-1 SD 1.1								
REQ-1 SD 1.2								
REQ-1 Class 1.1								
REQ-1 Class 1.2								
REQ-1 TC 1.1								
REQ-1 TC 1.2								

Keeping Traceability Links: The Process

Requirement ID	REQ-1 UC 1.1	REQ-1 UC 1.2	REQ-1 SD 1.1	REQ-1 SD 1.2	REQ-1 Class 1.1	REQ-1 Class 1.2	REQ-1 TC 1.1	REQ-1 TC 1.2
REQ-1 UC 1.1	X		X		X		X	
REQ-1 UC 1.2								
REQ-1 SD 1.1	X							
REQ-1 SD 1.2								
REQ-1 Class 1.1	X							
REQ-1 Class 1.2								
REQ-1 TC 1.1	X							
REQ-1 TC 1.2								

Keeping Traceability Links: The Process

Requirement ID	REQ-1 UC 1.1	REQ-1 UC 1.2	REQ-1 SD 1.1	REQ-1 SD 1.2	REQ-1 Class 1.1	REQ-1 Class 1.2	REQ-1 TC 1.1	REQ-1 TC 1.2
REQ-1 UC 1.1	X		X		X		X	
REQ-1 UC 1.2								
REQ-1 SD 1.1	X							
REQ-1 SD 1.2								
REQ-1 Class 1.1	X							
REQ-1 Class 1.2								
REQ-1 TC 1.1	X							
REQ-1 TC 1.2								

Keeping Traceability Links: The Process

Requirement ID	REQ-1 UC 1.1	REQ-1 UC 1.2	REQ-1 SD 1.1	REQ-1 SD 1.2	REQ-1 Class 1.1	REQ-1 Class 1.2	REQ-1 TC 1.1	REQ-1 TC 1.2
REQ-1 UC 1.1	X		X		X		X	
REQ-1 UC 1.2								
REQ-1 SD 1.1	X							
REQ-1 SD 1.2								
REQ-1 Class 1.1	X							
REQ-1 Class 1.2								
REQ-1 TC 1.1	X							
REQ-1 TC 1.2								

Forward Traceability Links

Keeping Traceability Links: The Process

Requirement ID	REQ-1 UC 1.1	REQ-1 UC 1.2	REQ-1 SD 1.1	REQ-1 SD 1.2	REQ-1 Class 1.1	REQ-1 Class 1.2	REQ-1 TC 1.1	REQ-1 TC 1.2
REQ-1 UC 1.1	X		X		X		X	
REQ-1 UC 1.2								
REQ-1 SD 1.1	X							
REQ-1 SD 1.2								
REQ-1 Class 1.1	X							
REQ-1 Class 1.2								
REQ-1 TC 1.1	X							
REQ-1 TC 1.2								

Backward Traceability Links

Keeping Traceability Links: Some Tools

FreeMind

http://freemind.sourceforge.net/wiki/index.php/Main_Page

Concordion

http://concordion.org

Keeping Traceability Links: Some Tools

FreeMind

http://freemind.sourceforge.net/wiki/index.php/Main_Page

Concordion

http://concordion.org

Google Sheets

Traceability can support several development activities, however…

Traceability can support several development activities, however…

Manual Process

Traceability can support several development activities, however…

Manual Process

Effort-prone task, especially because of software evolution

Traceability can support several development activities, however…

Manual Process

Effort-prone task, especially because of software evolution

Viewed by developers as a low priority task

Traceability can support several development activities, however…

Manual Process

Effort-prone task, especially because of software evolution

Viewed by developers as a low priority task

Most of the links are obvious

Traceability can support several development activities, however…

Manual Process

Effort-prone task, especially because of software evolution

Viewed by developers as a low priority task

Most of the links are obvious

So, is it really worth recording all the traceability links?

PART III
Traceability Dimensions

Traceability links can be classified in three ways

D1	D2	D3
Vertical	Structural	Implicit
Horizontal	Knowledge-based	Explicit

A. Bianchi, R. Fasolino, and G. Visaggio.
"An exploratory case study of the maintenance effectiveness of traceability models",
IEEE International Workshop on Program Comprehension, pp. 149-158, 2000.

Traceability links can be classified in three ways

DI

Vertical

Horizontal

The ability of tracing dependent artifacts within the same model

A. Bianchi, R. Fasolino, and G. Visaggio.
"An exploratory case study of the maintenance effectiveness of traceability models",
IEEE International Workshop on Program Comprehension, pp. 149-158, 2000.

Traceability links can be classified in three ways

DI

Vertical

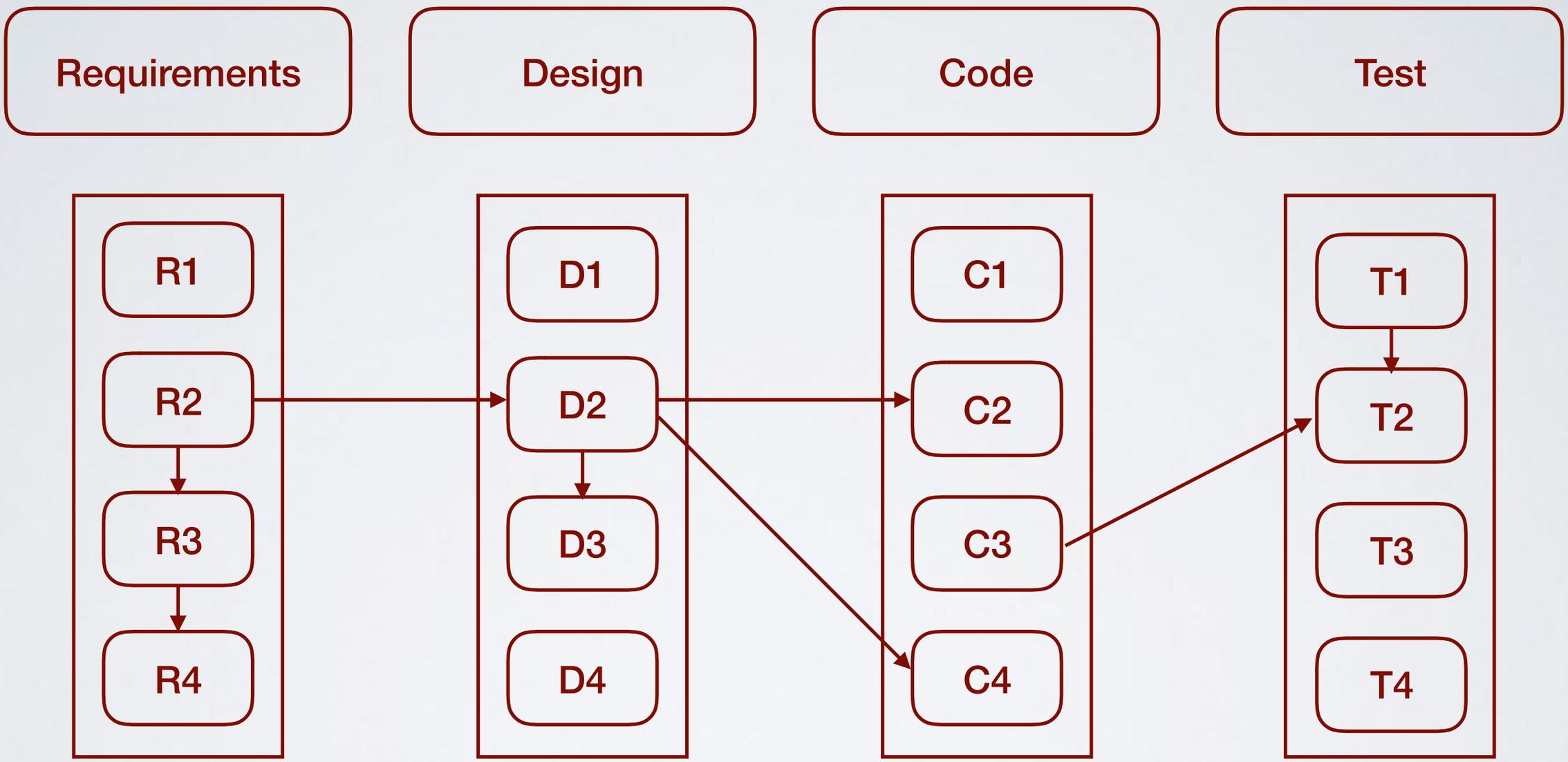
The ability of tracing dependent artifacts within the same model

Horizontal

The ability of tracing dependent artifacts between different models

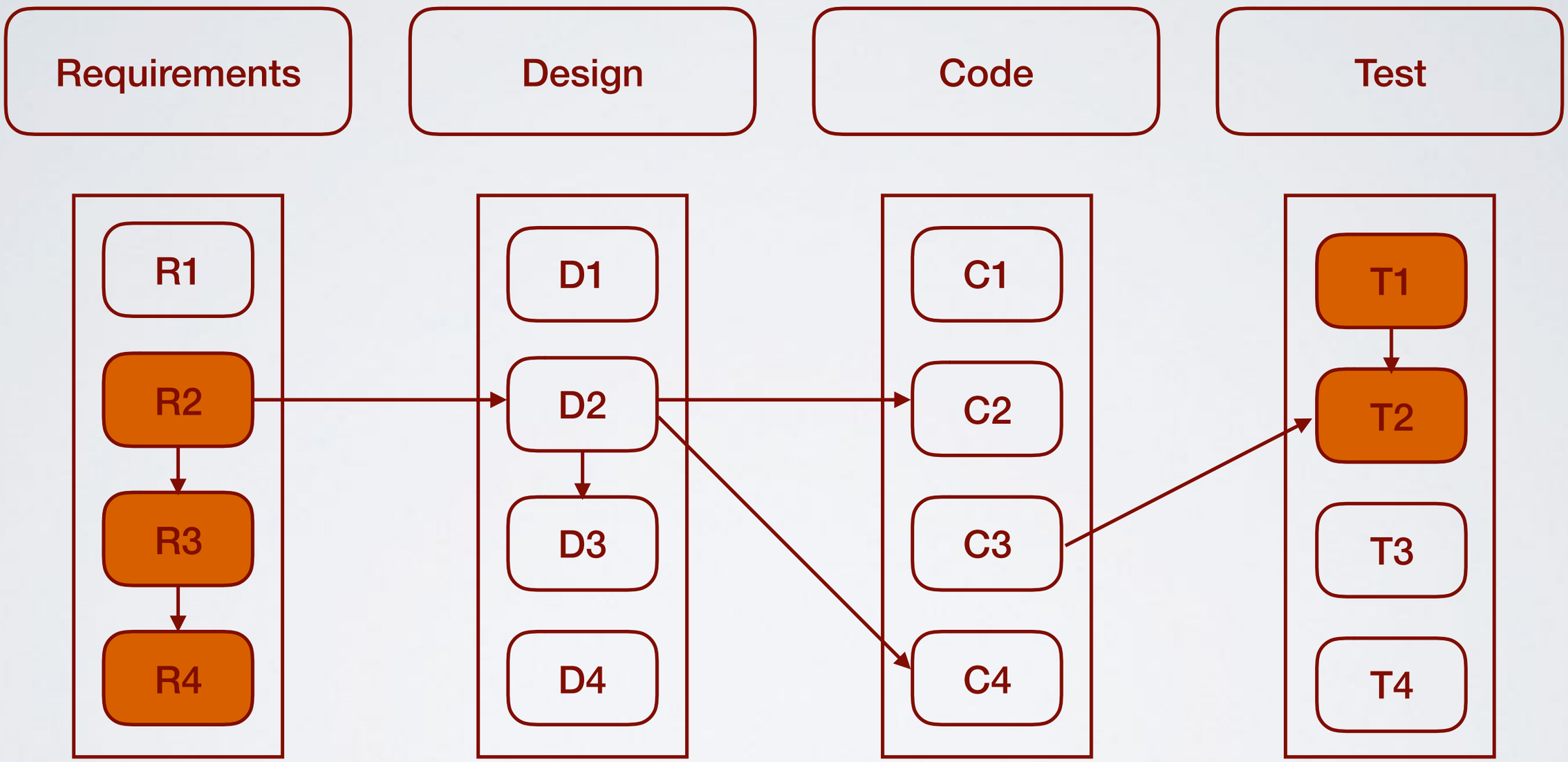
A. Bianchi, R. Fasolino, and G. Visaggio.
"An exploratory case study of the maintenance effectiveness of traceability models",
IEEE International Workshop on Program Comprehension, pp. 149-158, 2000.

Traceability links can be classified in three ways



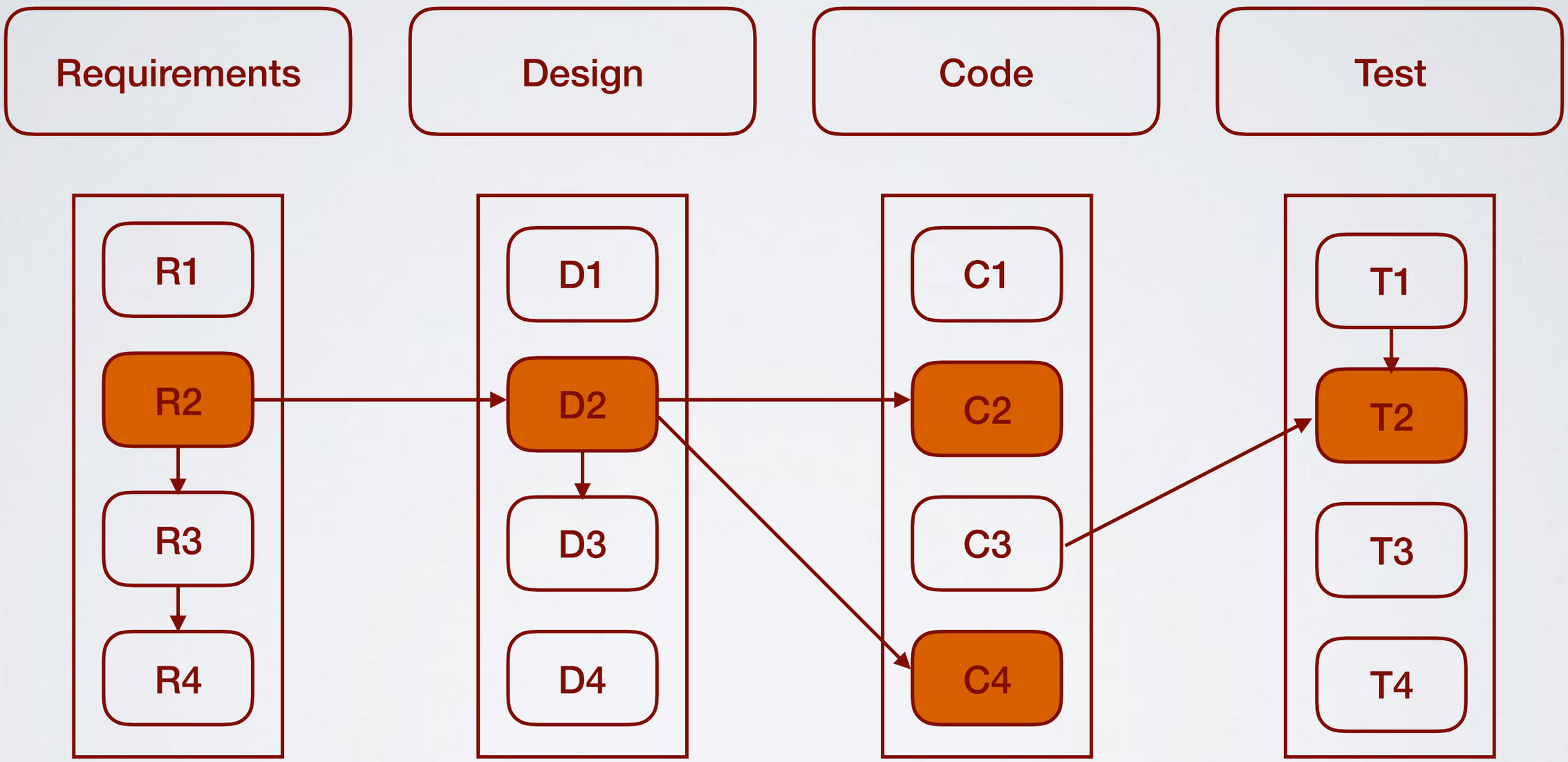
A. Bianchi, R. Fasolino, and G. Visaggio.
"An exploratory case study of the maintenance effectiveness of traceability models",
IEEE International Workshop on Program Comprehension, pp. 149-158, 2000.

Traceability links can be classified in three ways



A. Bianchi, R. Fasolino, and G. Visaggio.
"An exploratory case study of the maintenance effectiveness of traceability models",
IEEE International Workshop on Program Comprehension, pp. 149-158, 2000.

Traceability links can be classified in three ways



A. Bianchi, R. Fasolino, and G. Visaggio.
"An exploratory case study of the maintenance effectiveness of traceability models",
IEEE International Workshop on Program Comprehension, pp. 149-158, 2000.

Traceability links can be classified in three ways

D2

Structural

Traceability links that can be derived by analyzing the structure of artifacts

Knowledge-based

Traceability links whose discovery requires human interaction

A. Bianchi, R. Fasolino, and G. Visaggio.
"An exploratory case study of the maintenance effectiveness of traceability models",
IEEE International Workshop on Program Comprehension, pp. 149-158, 2000.

Traceability links can be classified in three ways

```
/* Insert a new user in the system.
 *  @param pUser: the user to insert.*/
public void insert(User pUser){

   connect = DBConnection.getConnection();

   String sql = "INSERT INTO USER"
      + “(login.first_name,last_name,password”
      + “,email.cell.id_parent) " + "VALUES ("
      + pUser.getLogin() + ","
      + pUser.getFirstName() + ","
      + pUser.getLastName() + ","
      + pUser.getPassword() + ","
      + pUser.getEMail() + ","
      + pUser.getCell() + ","
      + pUser.getIdParent() + ")";

   executeOperation(connect, sql);
}
```

```
/* Delete an user from the system.
 *  @param pUser: the user to delete.*/
public void delete(User pUser) {

   connect = DBConnection.getConnection();

   String sql = "DELETE FROM USER "
      + "WHERE id_user = "
      + pUser.getId();

   executeOperation(connect, sql);
}
```

Traceability links can be classified in three ways

```java
/* Insert a new user in the system.
 *  @param pUser: the user to insert.*/
public void insert(User pUser){

   connect = DBConnection.getConnection();

    String sql = "INSERT INTO USER"
       + “(login.first_name,last_name,password”
       + “,email.cell.id_parent) " + "VALUES ("
       + pUser.getLogin() + ","
       + pUser.getFirstName() + ","
       + pUser.getLastName() + ","
       + pUser.getPassword() + ","
       + pUser.getEMail() + ","
       + pUser.getCell() + ","
       + pUser.getIdParent() + ")";

   executeOperation(connect, sql);
}
```

Structural link with the DBConnection class

```java
/* Delete an user from the system.
 *  @param pUser: the user to delete.*/
public void delete(User pUser) {

   connect = DBConnection.getConnection();

    String sql = "DELETE FROM USER "
       + "WHERE id_user = "
       + pUser.getId();

   executeOperation(connect, sql);
 }
```

Traceability links can be classified in three ways

```
/* Insert a new user in the system.
 *  @param pUser: the user to insert.*/
public void insert(User pUser){

    connect = DBConnection.getConnection();

    String sql = "INSERT INTO USER"
        + “(login.first_name,last_name,password”
        + “,email.cell.id_parent) " + "VALUES ("
        + pUser.getLogin() + ","
        + pUser.getFirstName() + ","
        + pUser.getLastName() + ","
        + pUser.getPassword() + ","
        + pUser.getEMail() + ","
        + pUser.getCell() + ","
        + pUser.getIdParent() + ")";

    executeOperation(connect, sql);
}
```

The two methods are semantically related, but there is not a structural link between them

```
/* Delete an user from the system.
 *  @param pUser: the user to delete.*/
public void delete(User pUser) {

    connect = DBConnection.getConnection();

    String sql = "DELETE FROM USER "
        + "WHERE id_user = "
        + pUser.getId();

    executeOperation(connect, sql);
}
```

Traceability links can be classified in three ways

D3

Implicit

Traceability links are derived on-the-fly from software artifacts when needed

Explicit

Traceability links are stored in a repository

A. Bianchi, R. Fasolino, and G. Visaggio.
"An exploratory case study of the maintenance effectiveness of traceability models",
IEEE International Workshop on Program Comprehension, pp. 149-158, 2000.

So, is it really worth recording all the traceability links?

Useful to have a detailed control on the evolution of requirements

Useful to support management activities

Manual Process that need to be re-executed time-by-time

So, is it really worth recording all the traceability links?

Useful to have a detailed control on the evolution of requirements

Useful to support management activities

Manual Process that need to be re-executed time-by-time

Hybrid Approach

Only some type of links are explicitly stored

PART IV
Traceability Recovery

Suppose that you already developed the entire system without keeping traceability links

Suppose that you already developed the entire system without keeping traceability links

How can you recover them?

Suppose that you already developed the entire system without keeping traceability links

Analysis of structural relationships between classes

Recovering structural links

Heuristic-based approaches

Information Retrieval-based approaches

Recovering knowledge-based links

Suppose that you already developed the entire system without keeping traceability links

Analysis of structural relationships between classes

Recovering structural links

Suppose that you already developed the entire system without keeping traceability links

Analysis of structural relationships between classes

Recovering structural links

```
/* Insert a new user in the system.
 *  @param pUser: the user to insert.*/
public void insert(User pUser){

   connect = DBConnection.getConnection();
```

Suppose that you already developed the entire system without keeping traceability links

Heuristic-based approaches

Recovering knowledge-based links

Name-tracing: Regular expressions exploiting naming conventions, string edit distance

Suppose that you already developed the entire system without keeping traceability links

Heuristic-based approaches

Recovering knowledge-based links

Name-tracing: Regular expressions exploiting naming conventions, string edit distance

UC-AddUser

SD-AddUser

Class-AddUser

Suppose that you already developed the entire system without keeping traceability links

Heuristic-based approaches

Recovering knowledge-based links

Dynamic-analysis: Identifying source code elements involved during the execution of a scenario

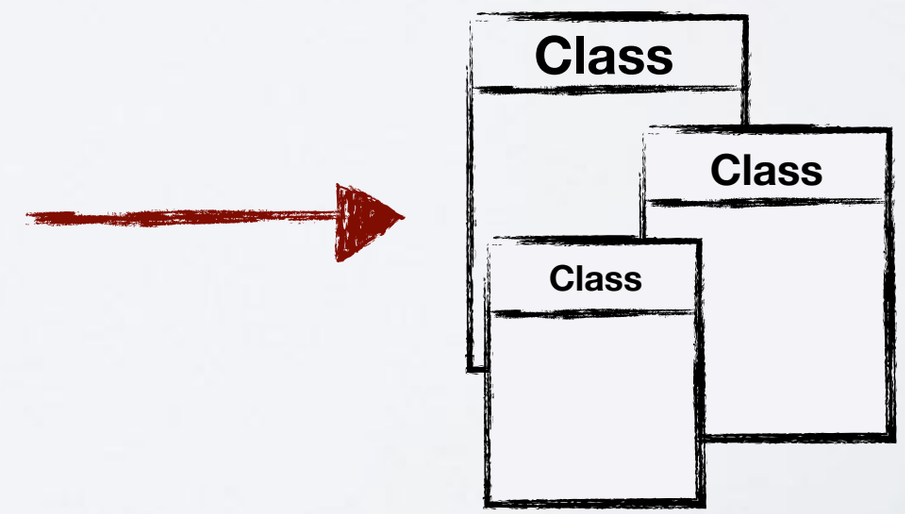
Suppose that you already developed the entire system without keeping traceability links

Heuristic-based approaches

Recovering knowledge-based links

Dynamic-analysis: Identifying source code elements involved during the execution of a scenario

Run Test Case



Suppose that you already developed the entire system without keeping traceability links

Heuristic-based approaches

Recovering knowledge-based links

Dynamic-analysis: Identifying source code elements involved during the execution of a scenario

When does this approach fail?

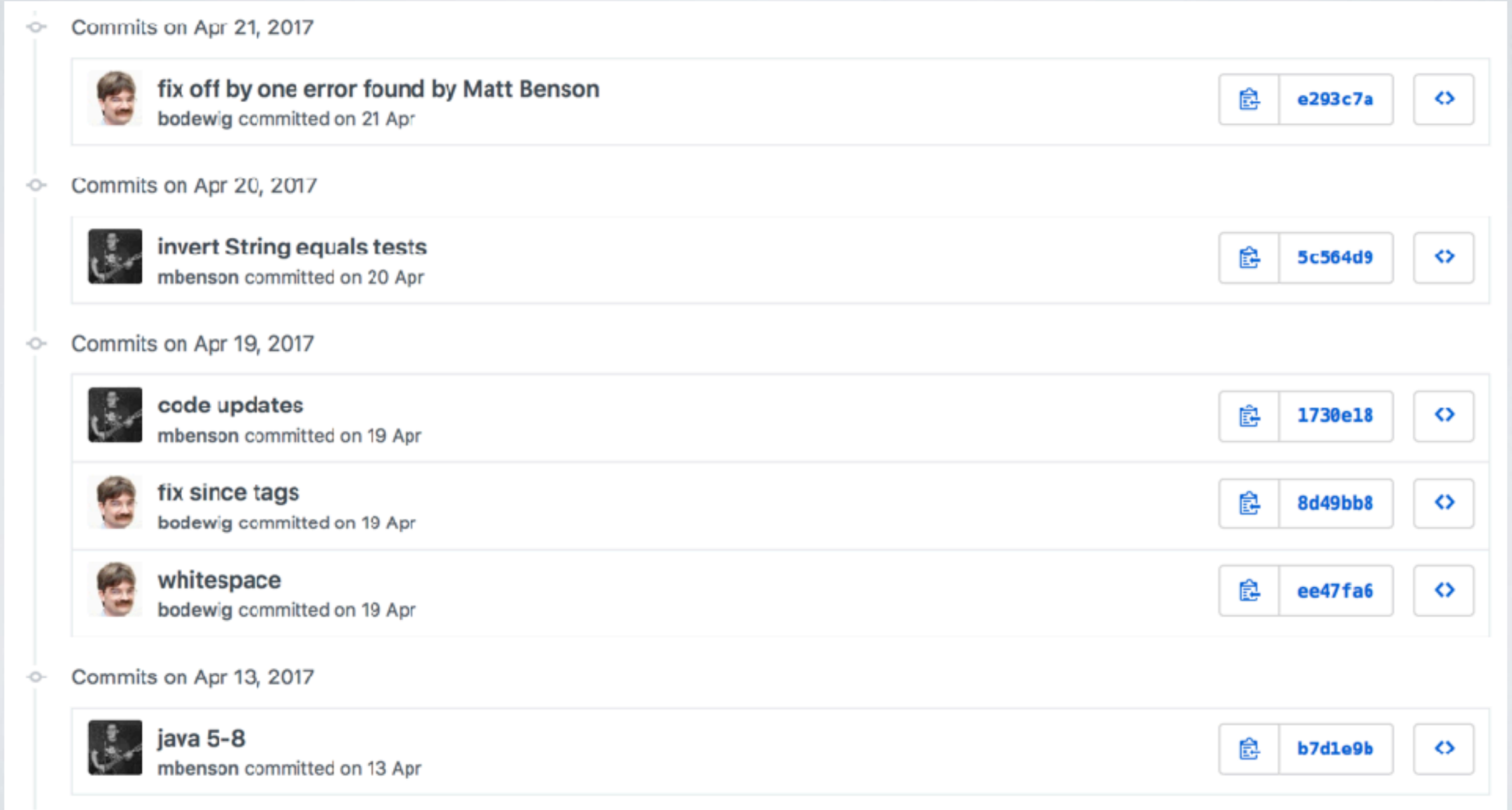
Suppose that you already developed the entire system without keeping traceability links

Heuristic-based approaches

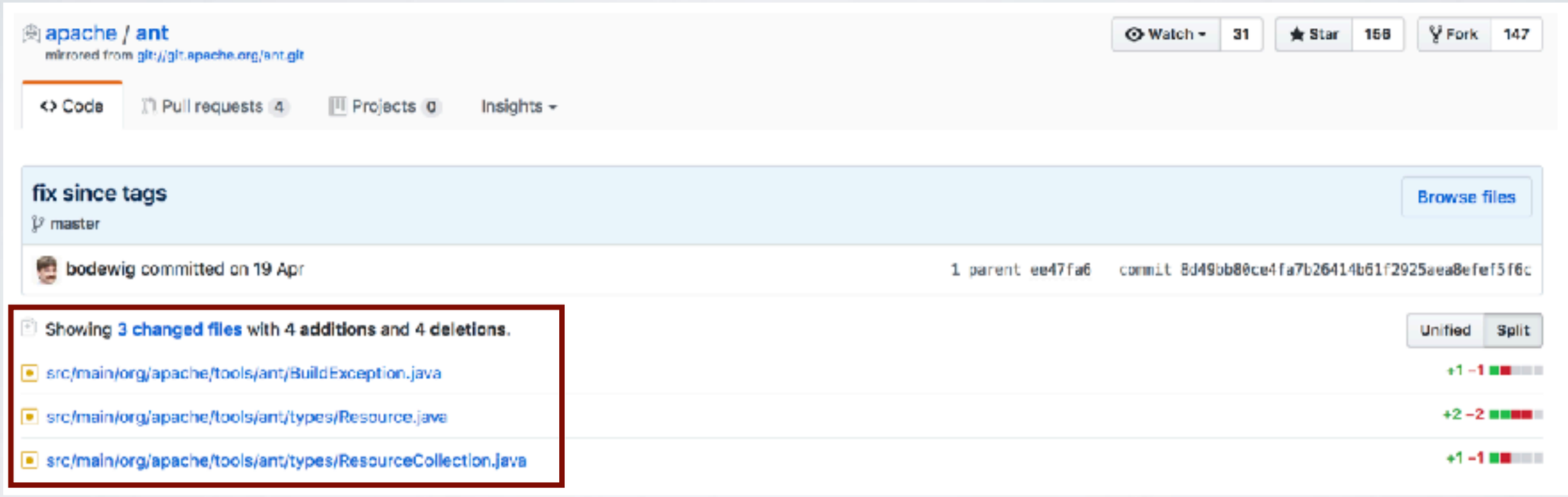
Recovering knowledge-based links

Mining Change Log: Identifying source code elements that frequently change together

Suppose that you already developed the entire system without keeping traceability links



Suppose that you already developed the entire system without keeping traceability links



The more two artifacts change together the higher the likelihood of the existence of a traceability link between them

Suppose that you already developed the entire system without keeping traceability links

Information Retrieval-based approaches

Recovering knowledge-based links

Suppose that you already developed the entire system without keeping traceability links

Information Retrieval-based approaches

Recovering knowledge-based links

Information Retrieval is the science of searching for information in a document

Suppose that you already developed the entire system without keeping traceability links

Information Retrieval-based approaches

Recovering knowledge-based links

Information Retrieval is the science of searching for information in a document

Often, it is done using textual information

Textual-based Traceability Recovery

Rationale: Documents and code contain text

Textual-based Traceability Recovery

Rationale: Documents and code contain text

```
/* Delete an user from the system.
 *  @param pUser: the user to
delete.*/
public void delete(User pUser) {

   connect =
DBConnection.getConnection();

   String sql = "DELETE FROM USER "
      + "WHERE id_user = "
      + pUser.getId();

   executeOperation(connect, sql);
 }
```

```
/* Delete an user from the system.
 *  @param pUser: the user to
delete.*/
public void delete(User pUser) {

   connect =
DBConnection.getConnection();

   String sql = "DELETE FROM USER "
      + "WHERE id_user = "
      + pUser.getId();

   executeOperation(connect, sql);
 }
```

Textual-based Traceability Recovery

The higher the textual similarity between two artifacts, the higher their similarity

```
/* Insert a new user in the system.
 *  @param pUser: the user to insert.*/
public void insert(User pUser){

   connect = DBConnection.getConnection();

   String sql = "INSERT INTO USER"
      + "(login,first_name,last_name,password"
      + ",email,cell,id_parent) " + "VALUES ("
      + pUser.getLogin() + ","
      + pUser.getFirstName() + ","
      + pUser.getLastName() + ","
      + pUser.getPassword() + ","
      + pUser.getEMail() + ","
      + pUser.getCell() + ","
      + pUser.getIdParent() + ")";

   executeOperation(connect, sql);
}
```

```
/* Delete an user from the system.
 *  @param pUser: the user to delete.*/
public void delete(User pUser) {

   connect = DBConnection.getConnection();

   String sql = "DELETE FROM USER "
      + "WHERE id_user = "
      + pUser.getId();

   executeOperation(connect, sql);
}
```

Textual-based Traceability Recovery

High textual similarity between two artifacts **likely mirrors a traceability link**

```
/* Insert a new user in the system.
 *  @param pUser: the user to insert.*/
public void insert(User pUser){

   connect = DBConnection.getConnection();

   String sql = "INSERT INTO USER"
      + "(login,first_name,last_name,password"
      + ",email,cell,id_parent) " + "VALUES ("
      + pUser.getLogin() + ","
      + pUser.getFirstName() + ","
      + pUser.getLastName() + ","
      + pUser.getPassword() + ","
      + pUser.getEMail() + ","
      + pUser.getCell() + ","
      + pUser.getIdParent() + ")";

   executeOperation(connect, sql);
}
```

```
/* Delete an user from the system.
 *  @param pUser: the user to delete.*/
public void delete(User pUser) {

   connect = DBConnection.getConnection();

   String sql = "DELETE FROM USER "
      + "WHERE id_user = "
      + pUser.getId();

   executeOperation(connect, sql);
}
```

The Process Extracting and Normalizing Text

```
private Connection connect = DBConnection.getConnection();
```

The Process Extracting and Normalizing Text

```
private Connection connect = DBConnection.getConnection();
```

Separating Composed Identifiers

```
private Connection connect = DB Connection.get Connection();
```

The Process Extracting and Normalizing Text

```
private Connection connect = DBConnection.getConnection();
```

Separating Composed Identifiers

```
private Connection connect = DB Connection.get Connection();
```

Lower Case Reduction

```
private connection connect = db connection.get connection();
```

The Process Extracting and Normalizing Text

```
private Connection connect = DBConnection.getConnection();
```

Separating Composed Identifiers

```
private Connection connect = DB Connection.get Connection();
```

Lower Case Reduction

```
private connection connect = db connection.get connection();
```

Removing Special Characters, programming keywords, and common English terms

```
connection connect = db connection get connection
```

The Process Extracting and Normalizing Text

```
private Connection connect = DBConnection.getConnection();
```

Separating Composed Identifiers

```
private Connection connect = DB Connection.get Connection();
```

Lower Case Reduction

```
private connection connect = db connection.get connection();
```

Removing Special Characters, programming keywords, and common English terms

```
connection connect = db connection get connection
```

Stemming

```
connect connect = db connect get connect
```

The Process

How to measure textual similarity?

The Process

How to measure textual similarity?

$$\text{Overlap index} = \text{terms}(a) \cap \text{terms}(b)$$

The Process

How to measure textual similarity?

Overlap index = terms(a) ∩ terms(b)

D1 = {A, B, C, E, F}

D2 = {A, C, D, E, G, H}

The Process

How to measure textual similarity?

Overlap index = terms(a) ∩ terms(b)

D1 = {A, B, C, E, F}

D2 = {A, C, D, E, G, H}

Overlap (D1,D2) = {A, C, E} = 3

How to measure textual similarity?

$$\text{Jaccard index} = \frac{\text{terms(a)} \cap \text{terms(b)}}{\text{terms(a)} \cup \text{terms(b)}}$$

The Process

How to measure textual similarity?

$$\text{Jaccard index} = \frac{\text{terms(a)} \cap \text{terms(b)}}{\text{terms(a)} \cup \text{terms(b)}}$$

D1 = {A, B, C, E, F}

D2 = {A, C, D, E, G, H}

The Process

How to measure textual similarity?

$$\text{Jaccard index} = \frac{\text{terms(a)} \cap \text{terms(b)}}{\text{terms(a)} \cup \text{terms(b)}}$$

$D1 = \{A, B, C, E, F\}$

$D2 = \{A, C, D, E, G, H\}$

$$\text{Jaccard}(D1,D2) = \frac{\{A, C, E\}}{\{A, B, C, D, E, F, G, O\}} = 0.38$$

How to measure textual similarity?

$$\text{Dice index} = \frac{\text{terms(a)} \cap \text{terms(b)}}{\max(\text{terms(a)}, \text{terms(b)})}$$

The Process

How to measure textual similarity?

$$\text{Dice index} = \frac{\text{terms(a)} \cap \text{terms(b)}}{\max(\text{terms(a)}, \text{terms(b)})}$$

D1 = {A, B, C, E, F}

D2 = {A, C, D, E, G, H}

The Process

How to measure textual similarity?

$$\text{Dice index} = \frac{\text{terms(a)} \cap \text{terms(b)}}{\max(\text{terms(a)}, \text{terms(b)})}$$

$D1 = \{A, B, C, E, F\}$

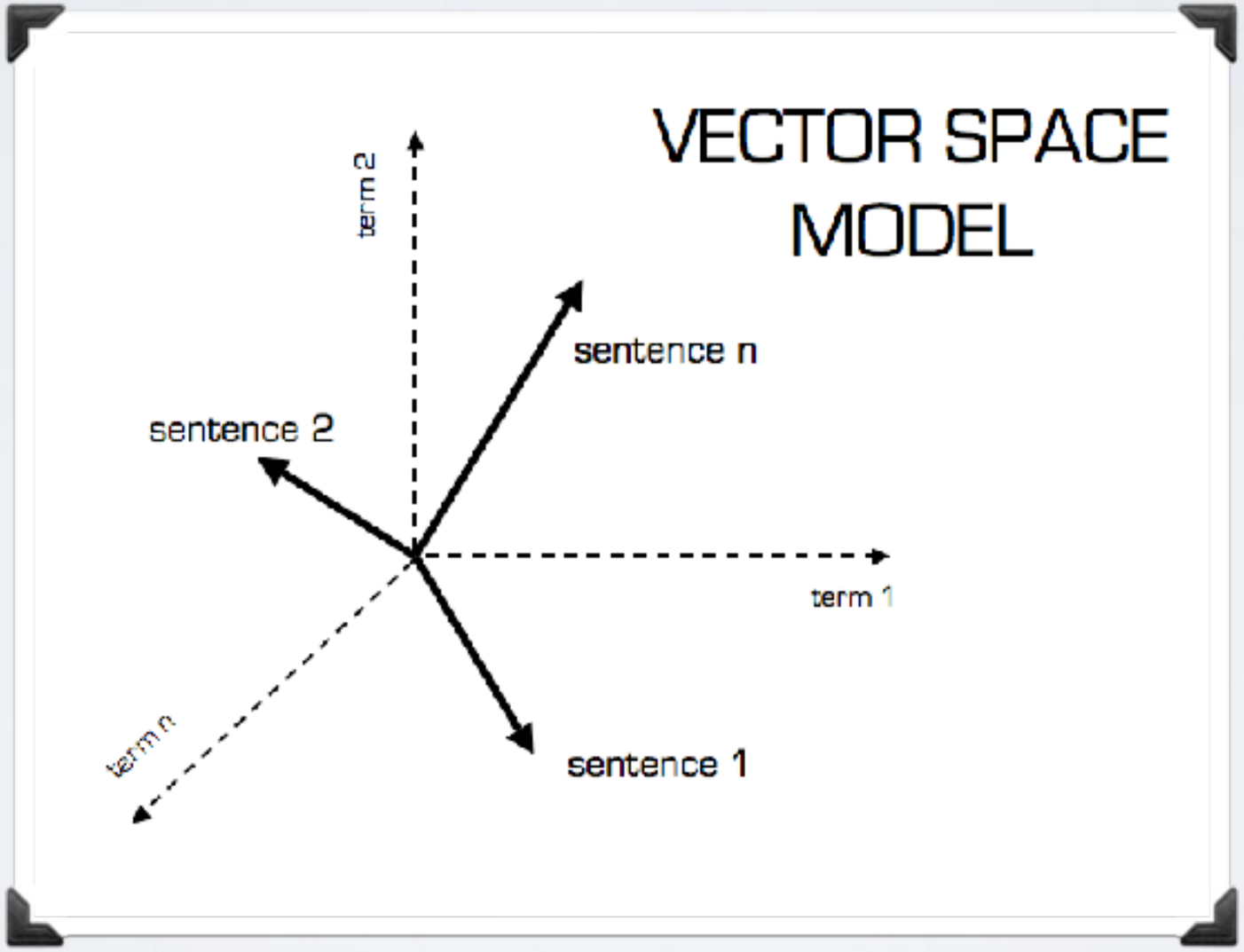
$D2 = \{A, C, D, E, G, H\}$

$$\text{Dice}(D1,D2) = \frac{\{A, C, E\}}{\{A, C, D, E, G, I\}} = 0.50$$

The Process

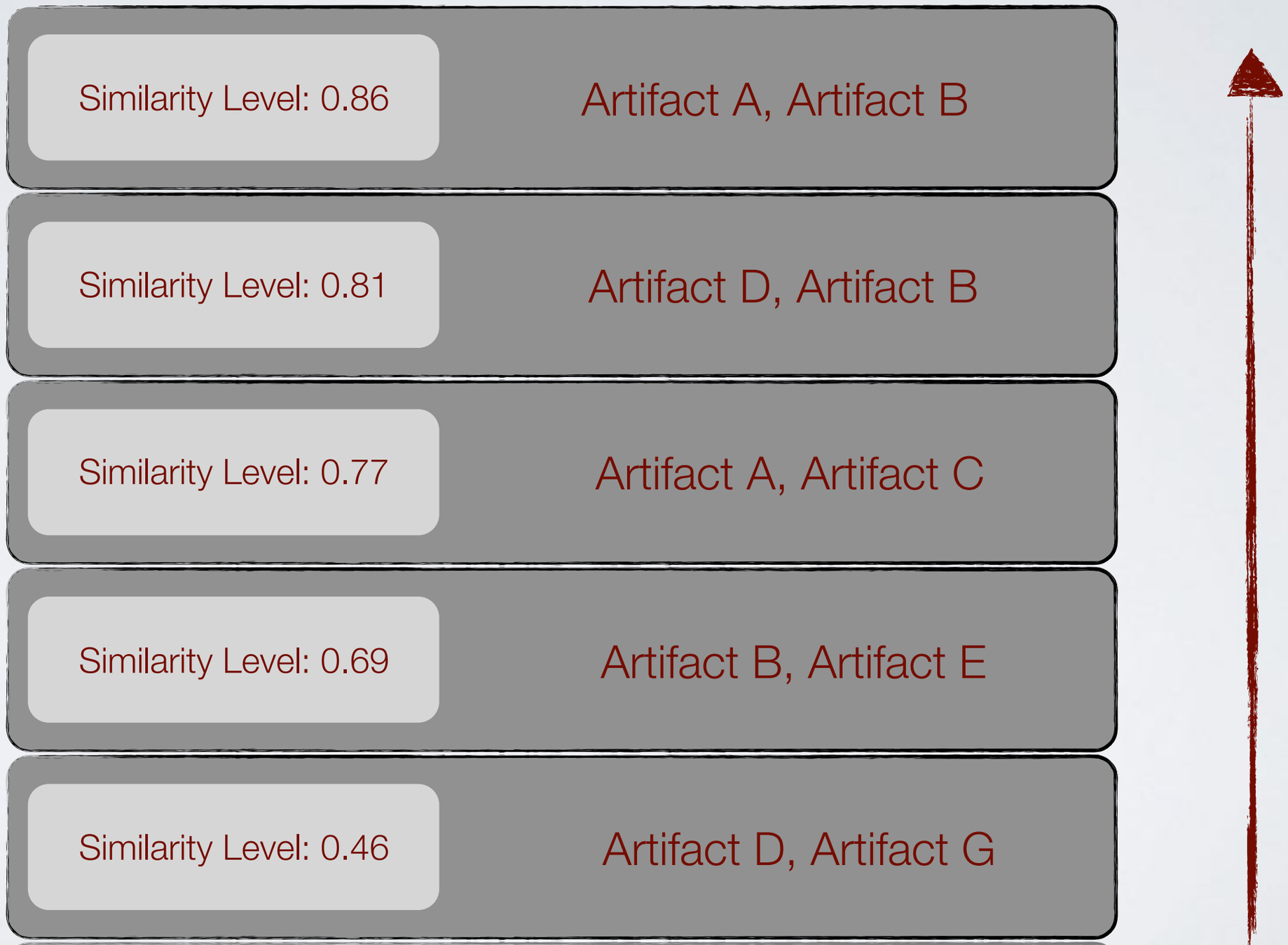
How to measure textual similarity?

There are also more complicated measures



The Process

The output of IR-based approaches: Ranked list



The Process

How to Evaluate an IR-based Traceability Recovery Approach?

The Process

How to Evaluate an IR-based Traceability Recovery Approach?

$$\text{precision} = \frac{\text{correct} \cap \text{retrieved}}{\text{retrieved}}$$

The Process

How to Evaluate an IR-based Traceability Recovery Approach?

$$\text{precision} = \frac{\text{correct} \cap \text{retrieved}}{\text{retrieved}}$$

precision = 0.67

The Process

How to Evaluate an IR-based Traceability Recovery Approach?

$$\text{recall} = \frac{\text{correct} \cap \text{retrieved}}{\text{correct}}$$

The Process

How to Evaluate an IR-based Traceability Recovery Approach?

$$\text{recall} = \frac{\text{correct} \cap \text{retrieved}}{\text{correct}}$$

Similarity Level: 0.86 — Artifact A, Artifact B ✓

Similarity Level: 0.81 — Artifact B, Artifact D ✓

Similarity Level: 0.77 — Artifact A, Artifact C ✗

Artifact A, Artifact C ✗

Artifact B, Artifact E ✗

precision = 0.67

recall = 0.50

The Process

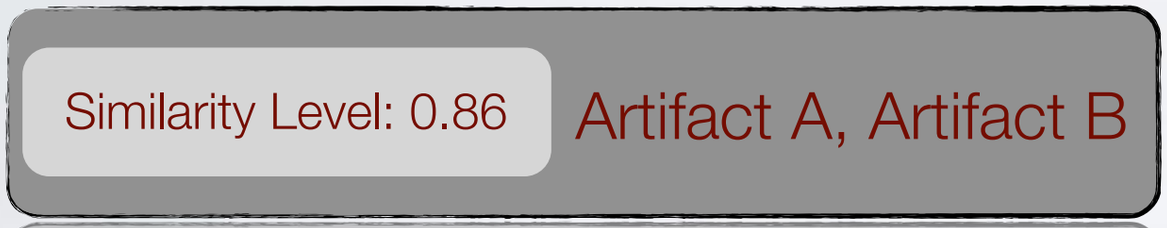
How to Evaluate an IR-based Traceability Recovery Approach?

Maximizing precision

The Process

How to Evaluate an IR-based Traceability Recovery Approach?

Maximizing precision: Just taking the traceability links for which you are sure about.



precision = 1.00

Artifact A, Artifact C ✗

Artifact B, Artifact E ✗

recall = 0.33

The Process

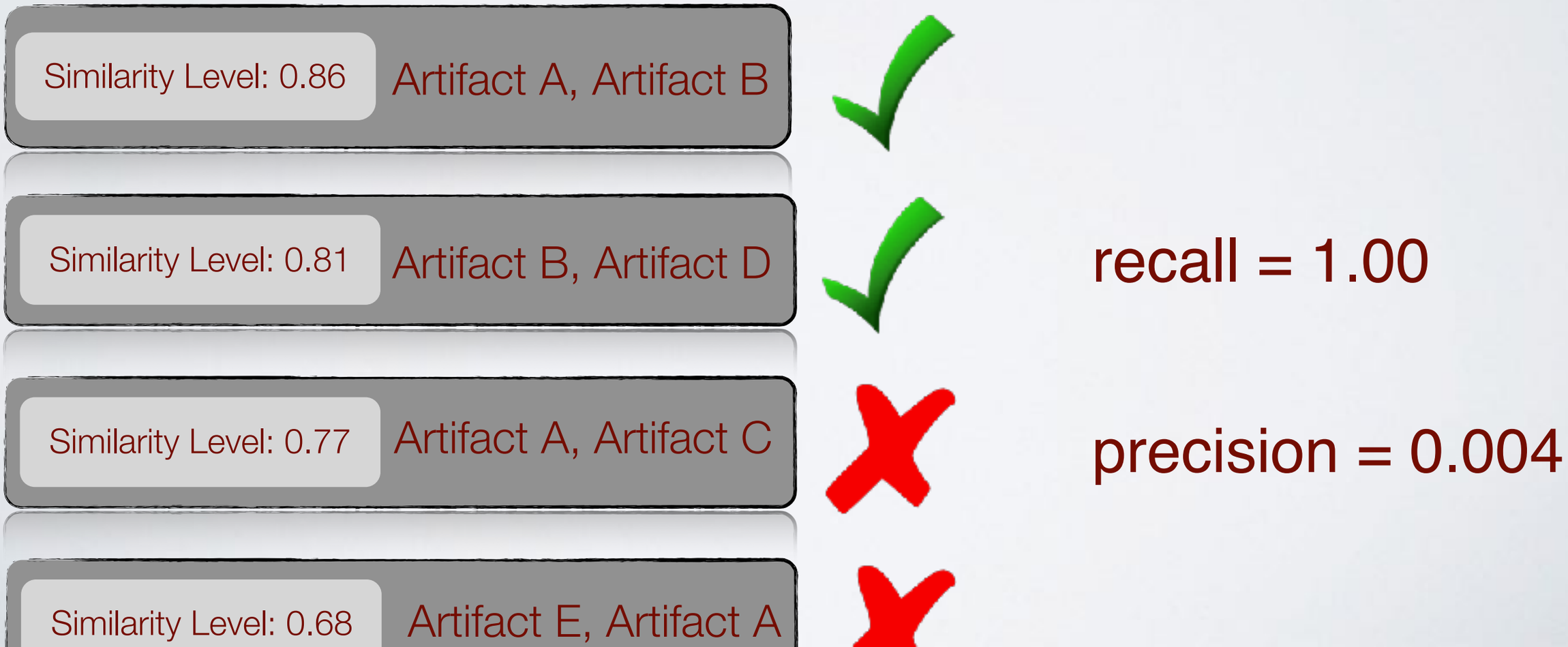
How to Evaluate an IR-based Traceability Recovery Approach?

Maximizing recall

The Process

How to Evaluate an IR-based Traceability Recovery Approach?

Maximizing recall: Just taking all the possible traceability links.



Precision and Recall - Pros and Cons

Maximizing both recall and precision is the goal, but it is not easy to achieve in practice

The Process Precision and Recall - Pros and Cons

Useful Measure to evaluate how an IR-based approach is good

They cannot be computed apriori!

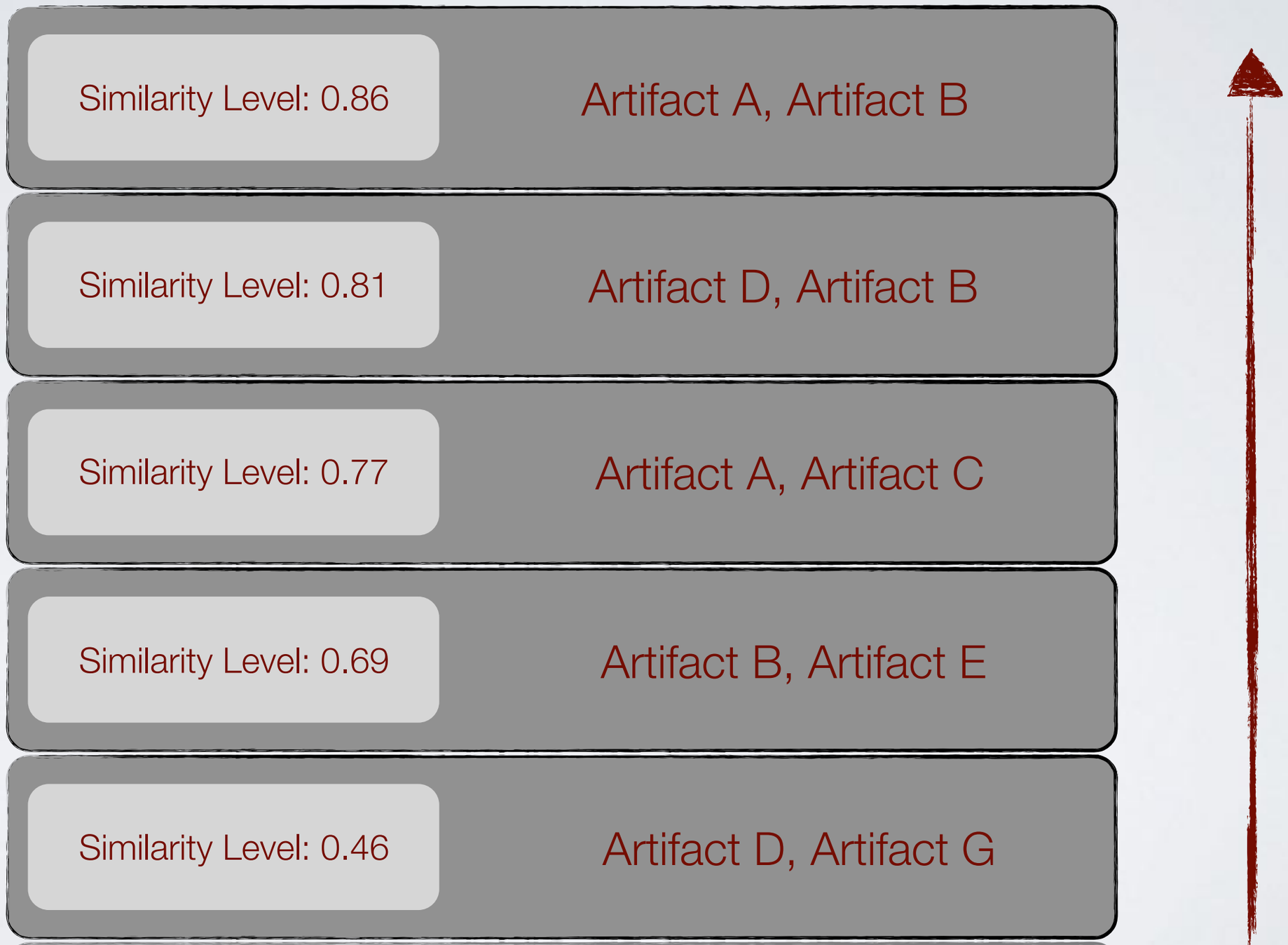
Precision and Recall - Pros and Cons

Useful Measure to evaluate how an IR-based approach is good

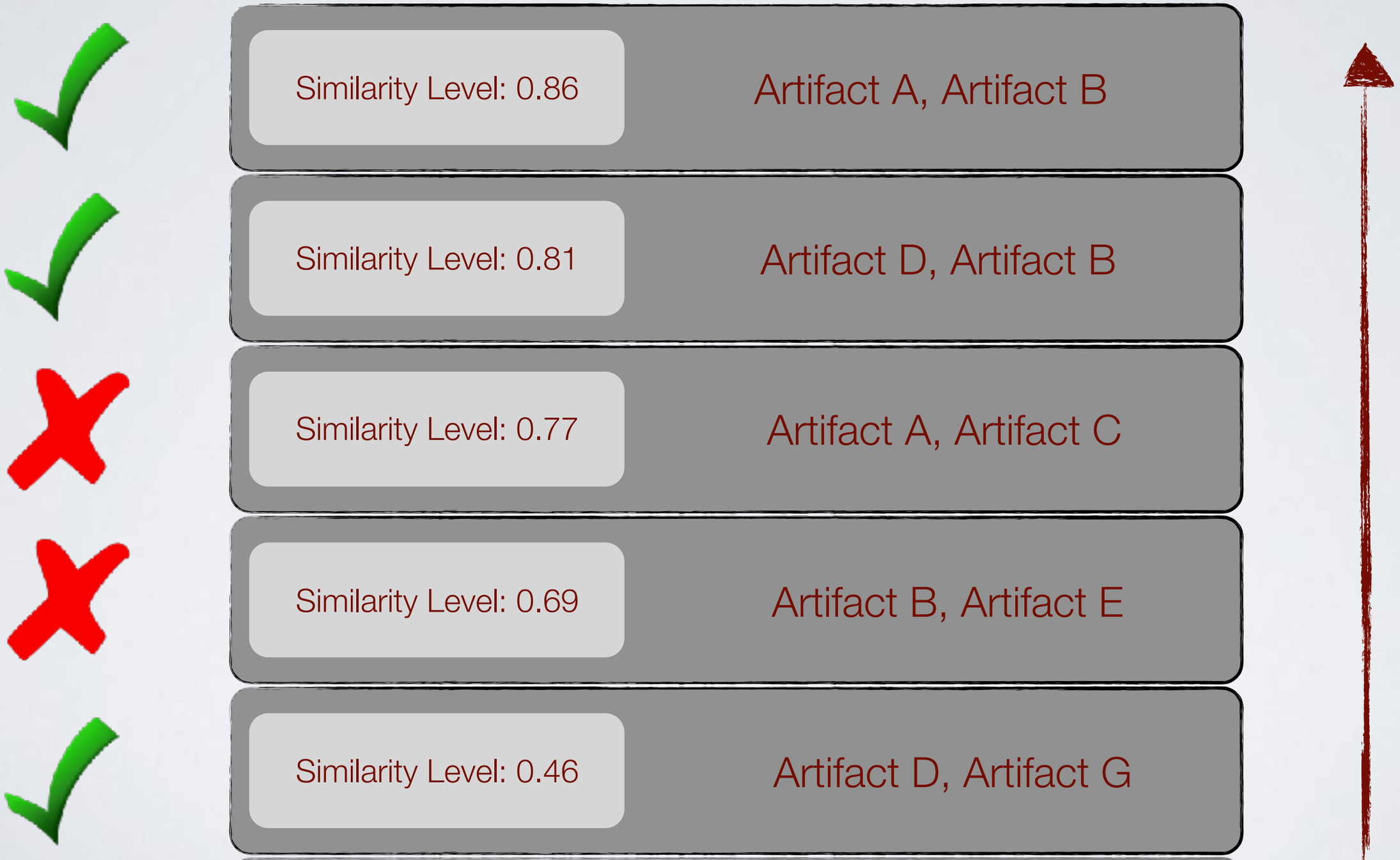
They cannot be computed apriori!

How to Cut a Ranked List in Practice?

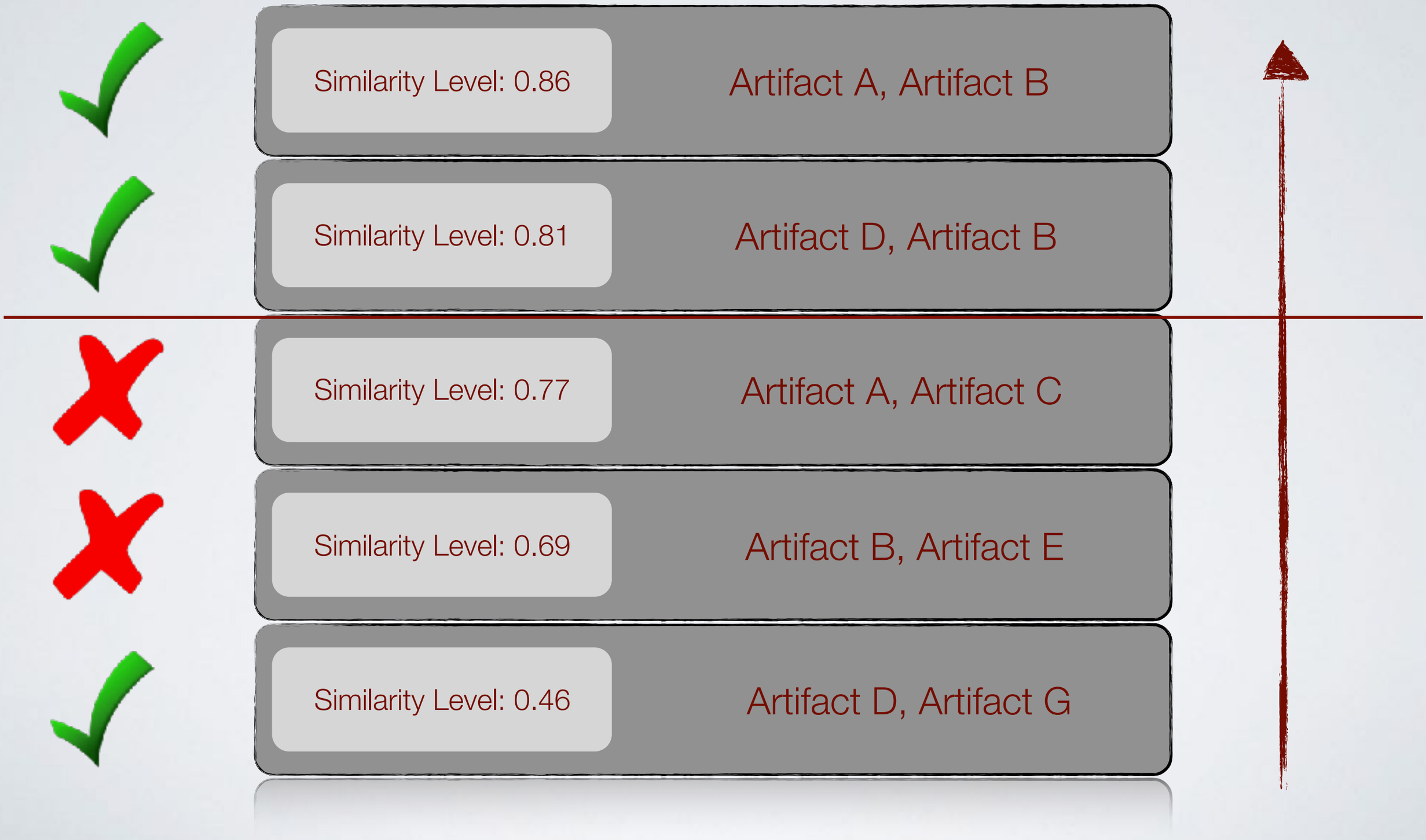
The Process How to cut the ranked list?



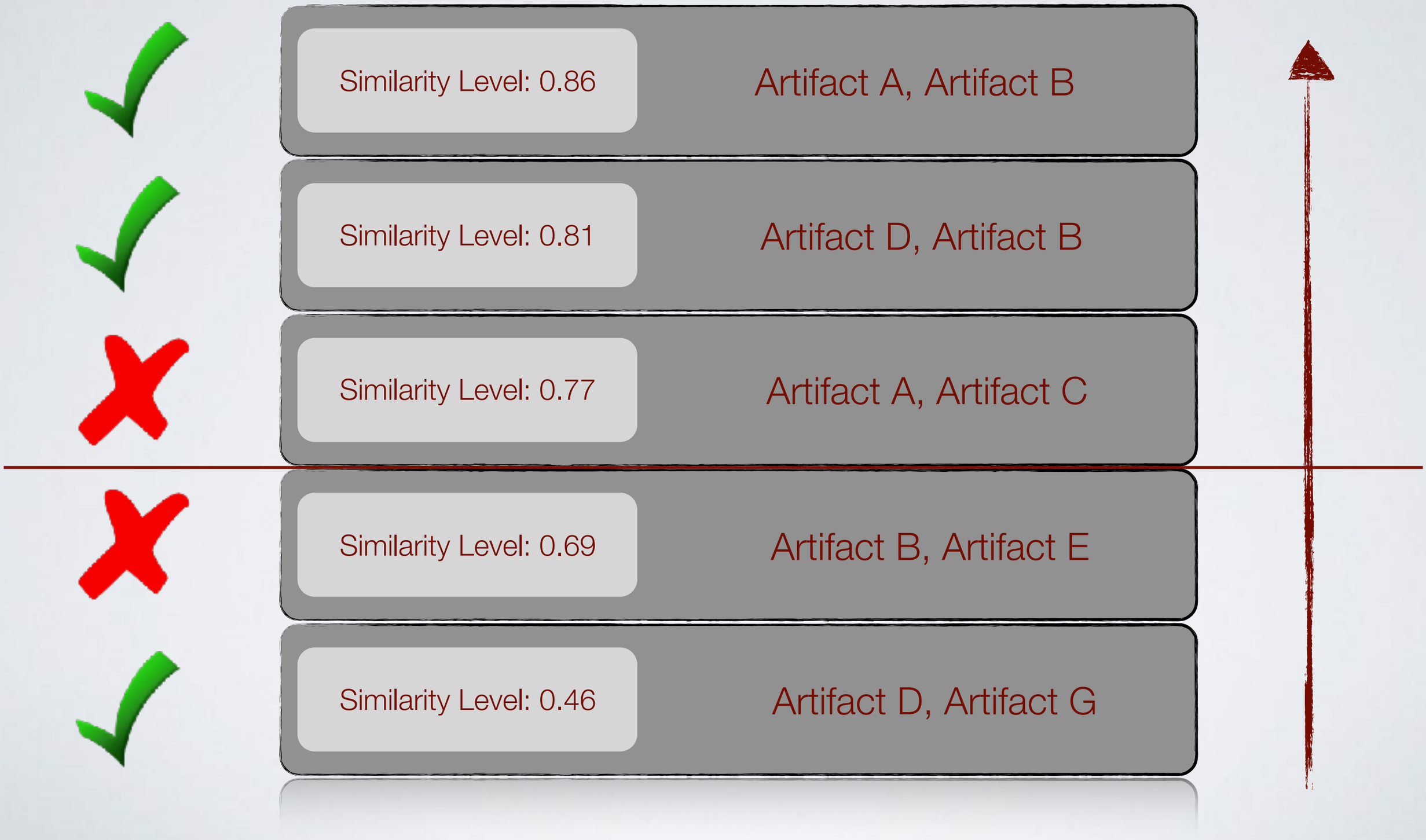
The Process How to cut the ranked list?



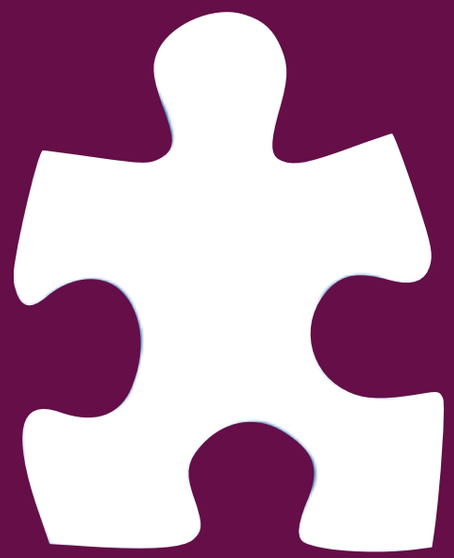
The Process Incremental Approach!



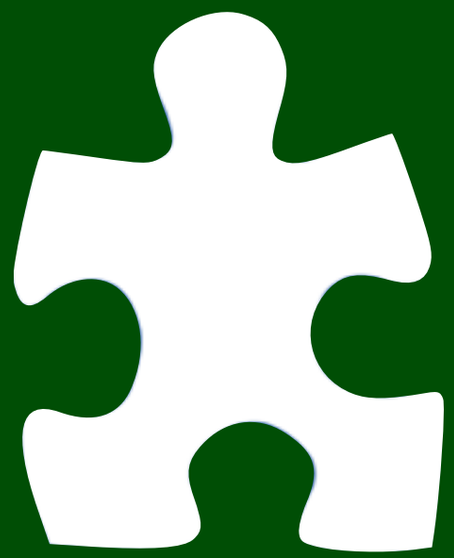
The Process Incremental Approach!



The Process Incremental Approach!



PART I
Understanding the Problem



PART II
Keeping Traceability Links

PART III
Traceability Dimensions

PART IV
Traceability Recovery

Related Literature

A. Bianchi, R. Fasolino, and G. Visaggio.
"An exploratory case study of the maintenance effectiveness of traceability models",
IEEE International Workshop on Program Comprehension, pp. 149-158, 2000.

G. Antoniol, G. Canfora, G. Casazza, A. De Lucia, and E. Merlo.
"Recovering Traceability Links between Code and Documentation",
IEEE Transactions on Software Engineering, pp. 970-983, 2002.

A. De Lucia, F. Fasano, R. Oliveto and G. Tortora.
"Recovering Traceability Links in Software Artifact Management Systems using Information Retrieval Methods",
ACM Transactions on Software Engineering and Methodology, vol. 16, num 4., 2007.

Managing Traceability and Traceability Recovery

Dr. Fabio Palomba

f.palomba@tudelft.nl, f.palomba@tue.nl

Software Engineering Methods
September 13rd, 2017