An Approach to Modeling and Supporting the Rework Process in Refactoring

Xiang Zhao, Leon Osterweil

Laboratory for Advanced Software Engineering Research
University of Massachusetts Amherst

Last Compiled: June 16, 2012
1 Rework in Processes
   - Overview
   - An Example of Rework
   - Questions during Rework
   - The Context in Rework

2 Refactoring as an Example of Rework
   - Overview
   - Modify Original Method Step Definition
   - Handle Unit Tests Failure Step Definition

3 An Approach to Supporting Rework
   - Overview
   - Data Derivation Graph
   - Prototype

4 Related Work

5 Future Work
Overview of Rework

Rework happens a lot and takes place very often in several domains like medical, election, and software development. However, it is not yet specifically addressed in most software engineering textbooks and guidelines.

People gain knowledge from rework. Intuitively, it can be understood as "going back to an earlier phase of the process?" Our understanding is that rework involves repeating activities that had been done previously in an earlier phase, but are now to be conducted in the context of a new phase guided by new contexts, history, and understandings.
Overview of Rework

- Rework happens a lot
  - Takes place very often in several domains like medical, election, and software development.
  - But not yet specifically addressed in most software engineering textbooks and guidelines.
Overview of Rework

- Rework happens a lot
  - Takes place very often in several domains like medical, election, and software development.
  - But not yet specifically addressed in most software engineering textbooks and guidelines.
- People gain knowledge from rework
Overview of Rework

- Rework happens a lot
  - Takes place very often in several domains like medical, election, and software development.
  - But not yet specifically addressed in most software engineering textbooks and guidelines.

- People gain knowledge from rework

- Intuitively: “Going back to an earlier phase of the process?”
Overview of Rework

- Rework happens a lot
  - Takes place very often in several domains like medical, election, and software development.
  - But not yet specifically addressed in most software engineering textbooks and guidelines.

- People gain knowledge from rework

- Intuitively: “Going back to an earlier phase of the process?”
  
  Our understanding: “Repeating activities that had been done previously in an earlier phase, but are now to be in the context of a new phase guided by new contexts, history, and understandings”
An Example of Rework

Xiang Zhao, Leon Osterweil

An Approach to Modeling and Supporting the Rework Process in Refactoring
An Example of Rework

Develop Software

→

Requirements Stage

↑ product

↓ state

Iteration

Design Stage

↓ state

↑ product

Implementation Stage

↓ state

↑ product

→

Requirements Stage

↓ state

↑ product

Design Stage

↓ state

↑ product
An Example of Rework

Xiang Zhao, Leon Osterweil

An Approach to Modeling and Supporting the Rework Process in Refactoring
Questions during Rework

- Where am I?
- What am I doing here?
- How did I get here?
- How did that work out?
- What alternatives do I have now?
- Which are likely to turn out best?
Questions during Rework

- Where am I?
- What am I doing here?
- How did I get here?
- How did that work out?
- What alternatives do I have now?
- Which are likely to turn out best?

The key is context
The Context in Rework

- What to expect in context: the collection of all information about previous and current process execution states
  - A set of artifact values
  - Text pointers (e.g. how deep in a recursion?)
  - Some plan for how to get back to “normal”
  - etc.
Refactoring as an Example of Rework

- Separate Query from Modifier
- Specify QueryModifier Method
- Modify Original Method
- Create Query
- Update References
- Deal with Original Method

- Process definitions based on [Fowler 1999]

Xiang Zhao, Leon Osterweil
An Approach to Modeling and Supporting the Rework Process in Refactoring
Refactoring as an Example of Rework

- Separate Query from Modifier
- Specify QueryModifier Method
- Modify Original Method
- Create Query
- Update References
- Deal with Original Method

Process definitions based on [Fowler 1999]
Modify Original Method Step Definition

- Make original method return a call to the new query
- Compile
- Run unit tests
- Modify Original Method
- Handle Compilation Error
- Handle Unit Tests Failure

- Different types of exception instances could be thrown and handled separately
Modify Original Method Step Definition

- Make original method return a call to the new query
- Compile
- Run unit tests
- Modify Original Method
- Handle Compilation Error
- Handle Unit Tests Failure

- UnitTestFailureException
- CompilationFailureException

Different types of exception instances could be thrown and handled separately.
Code refactoring contains multiple kinds of rework with arbitrary depth performed in different contexts.
Code refactoring contains multiple kinds of rework with arbitrary depth performed in different contexts

Should be guided by past history
An Approach to Supporting Rework

User working on refactoring

Agenda
- UnitTestFailure
- Check method call
- Check Query body

DDG
- Controls and Presents

Artifacts
- CheckingAccount.java
- Query Declaration
- Return Method Call

Juliette Interpreter
- Creates and Maintains

Refactoring Process Definition
- Monitors
An Approach to Supporting Rework

- Rigorous, executables process definition
  - Can provide articulate artifact flows and control flows.
  - Can use scoping to assure visibility of only appropriate artifacts

Xiang Zhao, Leon Osterweil
An Approach to Modeling and Supporting the Rework Process in Refactoring
An Approach to Supporting Rework

- Rigorous, executables process definition
  - Can provide articulate artifact flows and control flows.
  - Can use scoping to assure visibility of only appropriate artifacts
- Introspection
  - Ability to examine of current process state
An Approach to Supporting Rework

- Rigorous, executables process definition
  - Can provide articulate artifact flows and control flows.
  - Can use scoping to assure visibility of only appropriate artifacts

- Introspection
  - Ability to examine of current process state

- Retrospection
  - Ability to examine past process execution history
  - Data Derivation Graph

Xiang Zhao, Leon Osterweil
Defined templates for translating Little-JIL step executions into DAG fragments
Xiang Zhao, Leon Osterweil

An Approach to Modeling and Supporting the Rework Process in Refactoring

- Defined templates for translating Little-JIL step executions into DAG fragments
- Basic Features
  - Represents how artifacts are derived from each other
  - Incorporates scoping, nesting, hierarchy information
Defined templates for translating Little-JIL step executions into DAG fragments

**Basic Features**

- Represents how artifacts are derived from each other
- Incorporates scoping, nesting, hierarchy information

**Additional Features**

- Links to previous artifacts values
- Detailed history is inferable

Xiang Zhao, Leon Osterweil

An Approach to Modeling and Supporting the Rework Process in Refactoring
- Defined templates for translating Little-JIL step executions into DAG fragments

- Basic Features
  - Represents how artifacts are derived from each other
  - Incorporates scoping, nesting, hierarchy information

- Additional Features
  - Links to previous artifacts values
  - Detailed history is inferable

- Can generate DDGs dynamically while the process is executing
Prototype

Xiang Zhao, Leon Osterweil

An Approach to Modeling and Supporting the Rework Process in Refactoring
Prototype
Related Work

- Refactoring is widely supported in modern IDEs, few of which care about contexts.
- Mylyn[Kersten et al. AOSD ’05] stresses contexts, but a rigorous process model is needed to help to understand the artifacts.
- Data Provenance: VisTrails[Callahan et al. SIGMOD ’06], Kepler[Altintas et al. SSDBM ’04], and etc.
Future Work

- Ripple effects
- Study more refactoring patterns
- Persist DDG for Query
- Application of Speculative Analysis (Prospection)
Thank You

- Questions?