Continuous Recommendations for Repairing Robustness Anomalies

Eiji Adachi Barbosa
Alessandro Garcia

LES | DI |PUC-Rio - Brazil

OPUS Research Group

Exception Handling

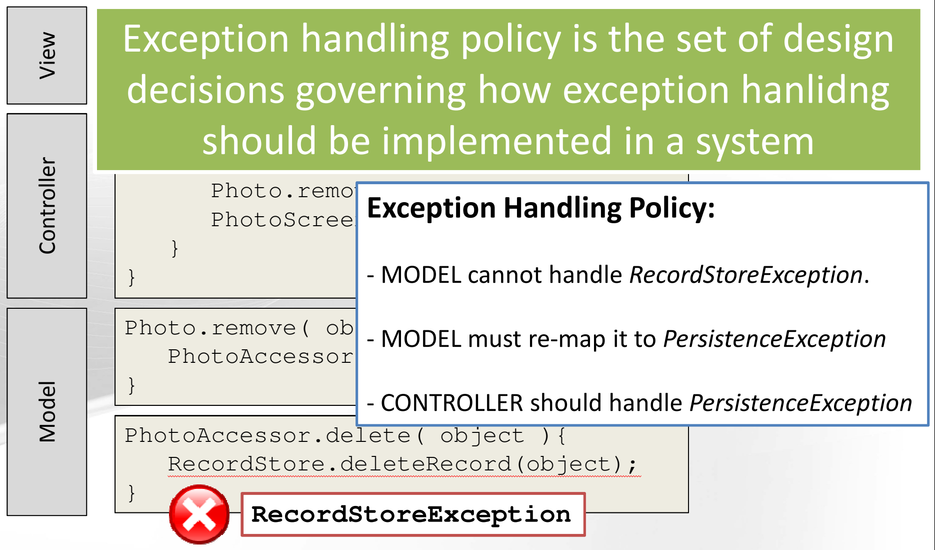
- Process of signaling exceptions upon the detection of runtime errors and taking actions to respond to their occurrence
  - Exception handling is central to robust software development
- Exception handling code is often the least documented, tested and understood part of a system [1]
- Faults per lines of code in exception handling code is three times higher than in normal code [2]
  - In some cases, exception handling code can concentrate approximately 60% of faults in a system [1]

[1] Cristian, F. "Exception handling and software fault tolerance." 1982.
[2] Sawadpong, P., Allen, E. B., & Williams, B. J. Exception handling defects: An empirical study. 2012.

Lack of Explicit Exception Handling Policies

Problem Statement

Exception handling policy is the set of design decisions governing how exception hanlidng should be implemented in a system



```
        Photo.remo
        PhotoScree
    }
}
```

```
Photo.remove( ob
    PhotoAccessor
}
```

```
PhotoAccessor.delete( object ){
    RecordStore.deleteRecord(object);
}
```

RecordStoreException

Exception Handling Policy:

- MODEL cannot handle *RecordStoreException*.

- MODEL must re-map it to *PersistenceException*

- CONTROLLER should handle *PersistenceException*

Lack of Explicit Exception Handling Policies

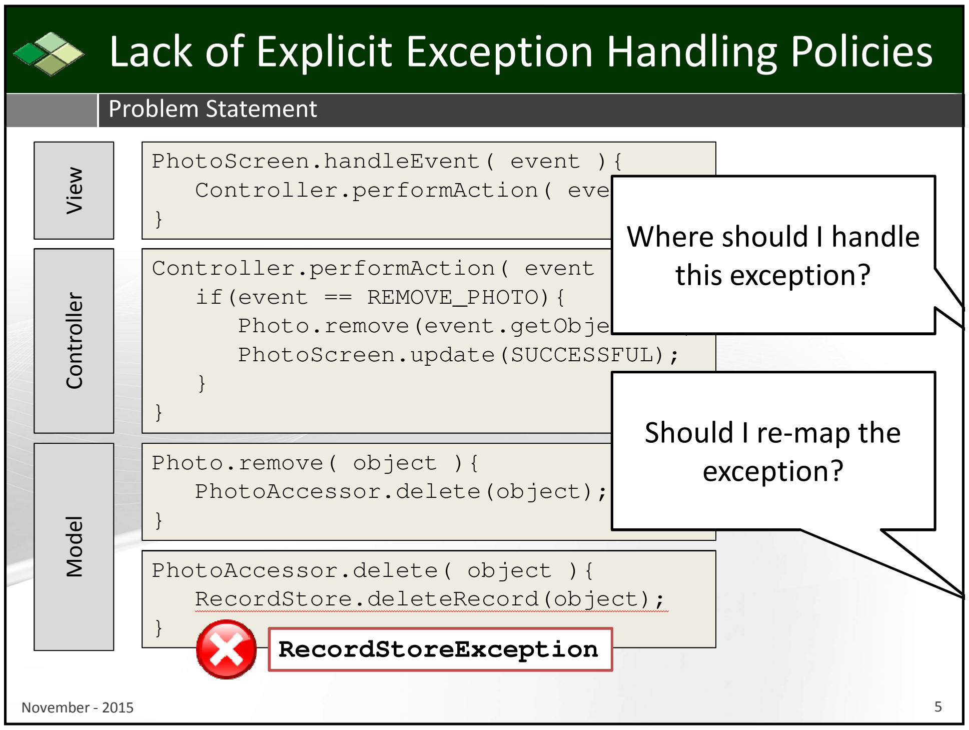
Problem Statement

- Not much effort is spent in documenting exception handling policies [3]
- In an empirical study we conducted, developers pinpointed implicit policies in their systems:

(Container) should throw the correct exception if an application attempts to modify the associated JNDI context

execute() cannot throw JasperException, so it matches the signature for Task.execute()

[3] Ebert, F., Castor, F., Serebrenik, A. An exploratory study on exception handling bugs in Java programs. 2015.



Lack of Explicit Exception Handling Policies

Problem Statement

View
Controller
Model
Where should I handle this exception?
Should I re-map the exception?
RecordStoreException

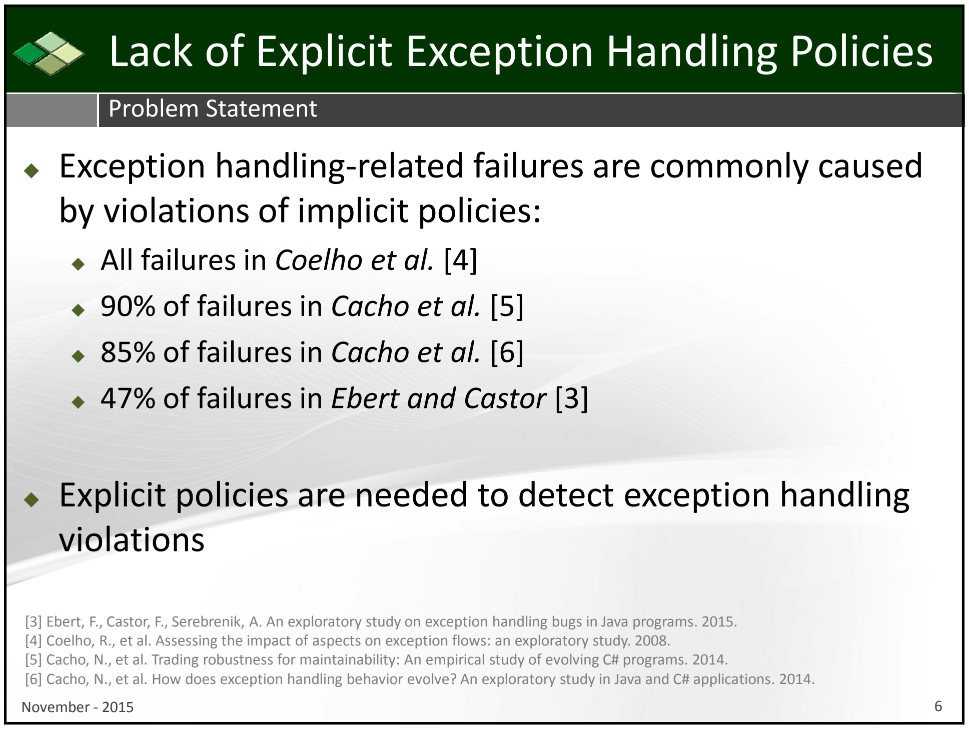
```
PhotoScreen.handleEvent( event ){
   Controller.performAction( eve
}

Controller.performAction( event
   if(event == REMOVE_PHOTO){
      Photo.remove(event.getObje
      PhotoScreen.update(SUCCESSFUL);
   }
}

Photo.remove( object ){
   PhotoAccessor.delete(object);
}

PhotoAccessor.delete( object ){
   RecordStore.deleteRecord(object);
}
```

November - 2015 5



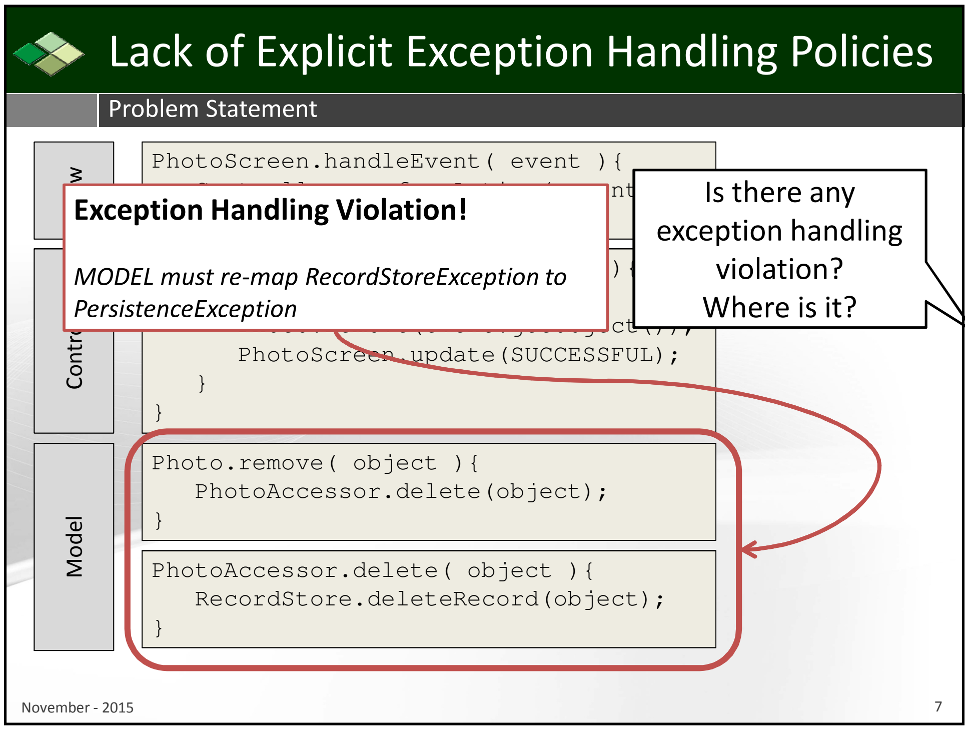
Lack of Explicit Exception Handling Policies

Problem Statement

- Exception handling-related failures are commonly caused by violations of implicit policies:
  - All failures in *Coelho et al.* [4]
  - 90% of failures in *Cacho et al.* [5]
  - 85% of failures in *Cacho et al.* [6]
  - 47% of failures in *Ebert and Castor* [3]
- Explicit policies are needed to detect exception handling violations

[3] Ebert, F., Castor, F., Serebrenik, A. An exploratory study on exception handling bugs in Java programs. 2015.
[4] Coelho, R., et al. Assessing the impact of aspects on exception flows: an exploratory study. 2008.
[5] Cacho, N., et al. Trading robustness for maintainability: An empirical study of evolving C# programs. 2014.
[6] Cacho, N., et al. How does exception handling behavior evolve? An exploratory study in Java and C# applications. 2014.

November - 2015 6



Lack of Explicit Exception Handling Policies

Problem Statement

View | Control | Model

```
PhotoScreen.handleEvent( event ){
        PhotoScreen.update(SUCCESSFUL);
    }
}
```

```
Photo.remove( object ){
   PhotoAccessor.delete(object);
}
```

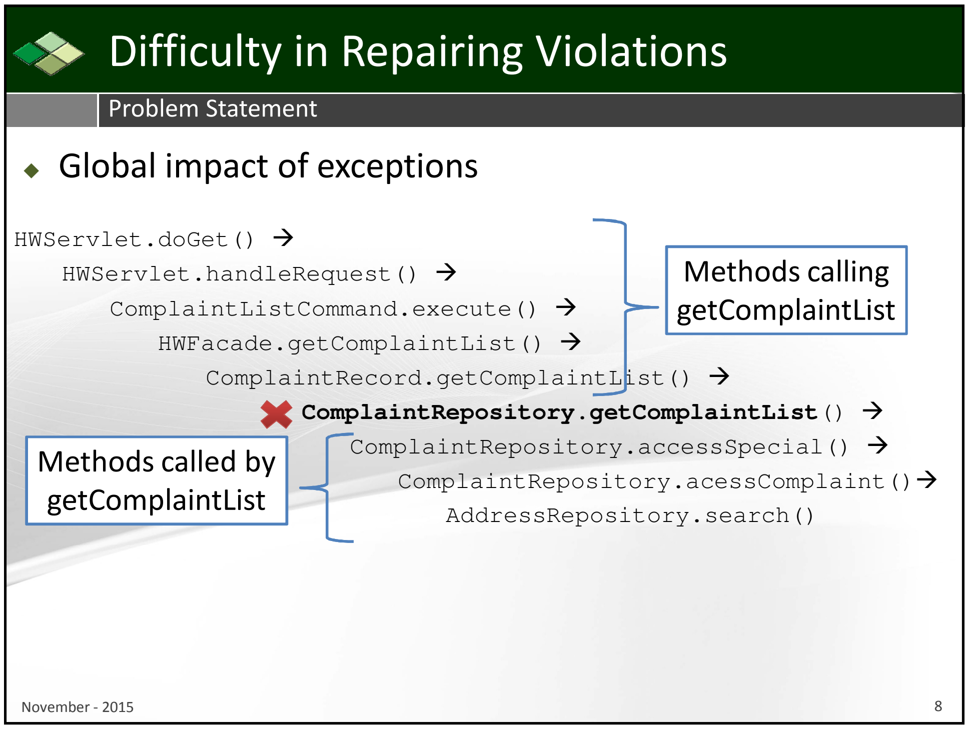
```
PhotoAccessor.delete( object ){
   RecordStore.deleteRecord(object);
}
```

Exception Handling Violation!

MODEL must re-map RecordStoreException to PersistenceException

Is there any exception handling violation? Where is it?

November - 2015 7



Difficulty in Repairing Violations

Problem Statement

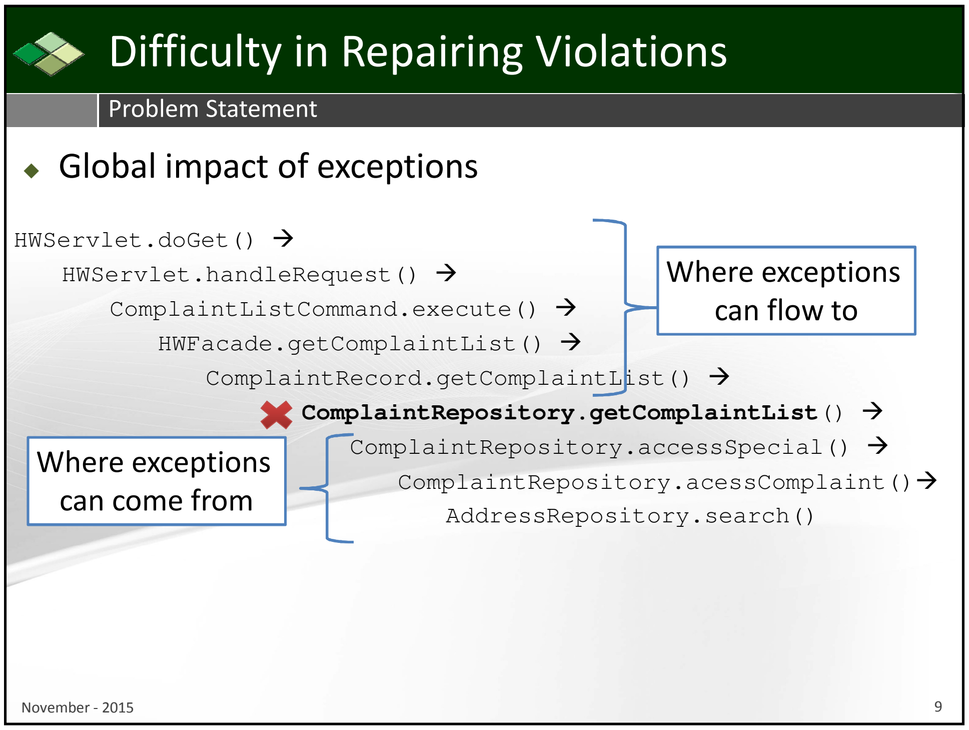
- Global impact of exceptions

```
HWServlet.doGet() →
    HWServlet.handleRequest() →
        ComplaintListCommand.execute() →
            HWFacade.getComplaintList() →
                ComplaintRecord.getComplaintList() →
                    ComplaintRepository.getComplaintList() →
                        ComplaintRepository.accessSpecial() →
                            ComplaintRepository.acessComplaint()→
                                AddressRepository.search()
```

Methods calling getComplaintList

Methods called by getComplaintList

November - 2015 8



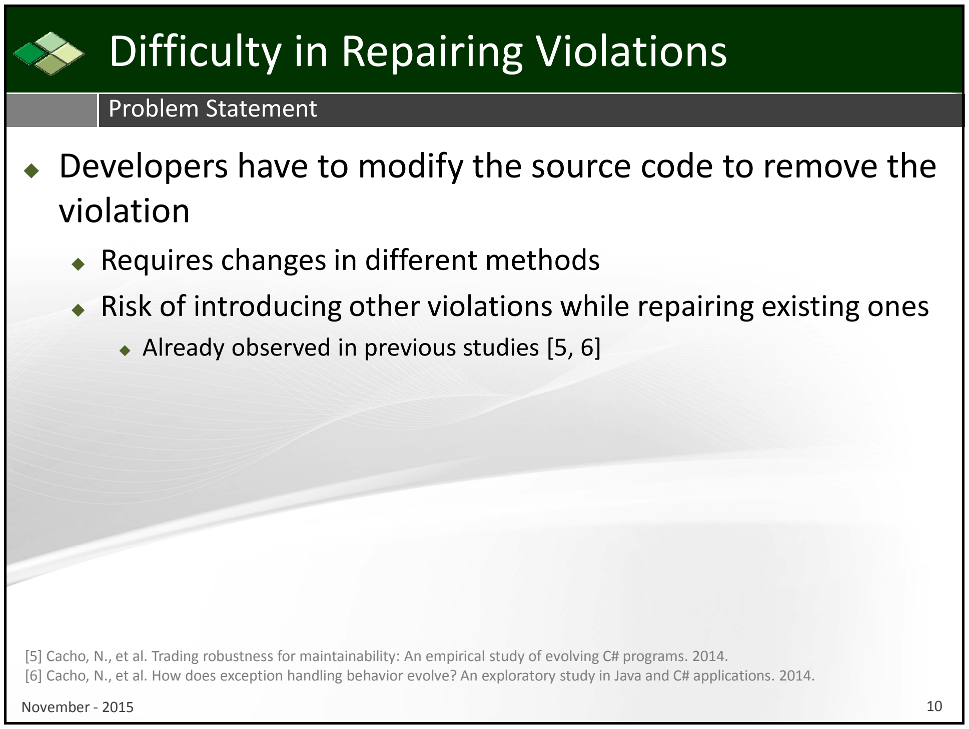
Difficulty in Repairing Violations

Problem Statement

- Global impact of exceptions

```
HWServlet.doGet() →
    HWServlet.handleRequest() →
        ComplaintListCommand.execute() →
            HWFacade.getComplaintList() →
                ComplaintRecord.getComplaintList() →
                    ComplaintRepository.getComplaintList() →
                        ComplaintRepository.accessSpecial() →
                            ComplaintRepository.acessComplaint() →
                                AddressRepository.search()
```

November - 2015 9



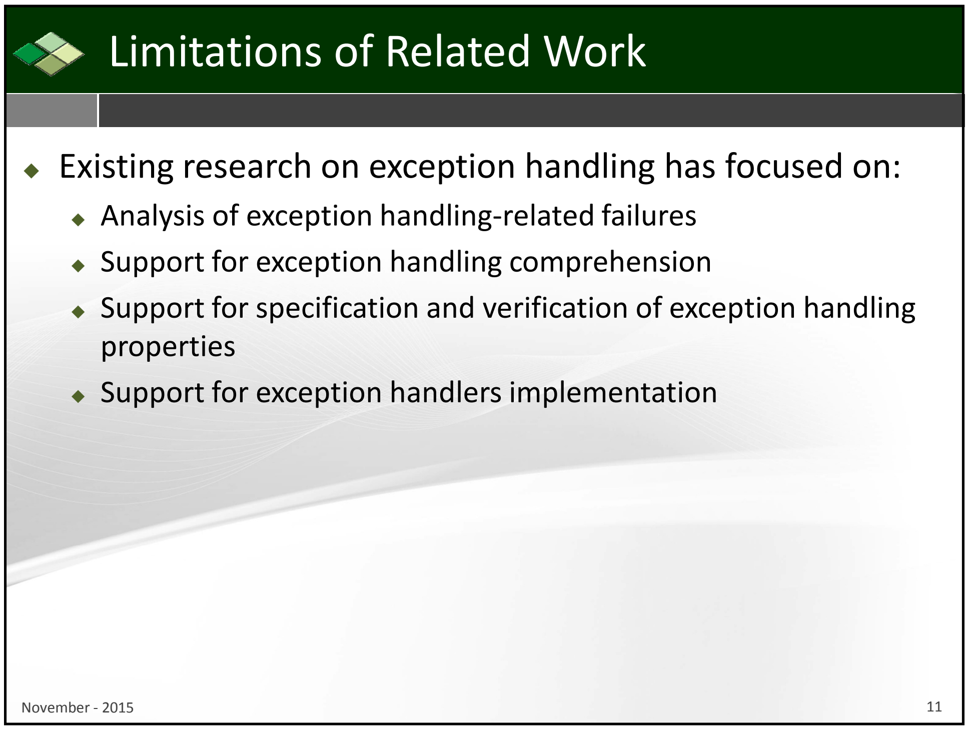
Difficulty in Repairing Violations

Problem Statement

- Developers have to modify the source code to remove the violation
  - Requires changes in different methods
  - Risk of introducing other violations while repairing existing ones
    - Already observed in previous studies [5, 6]

[5] Cacho, N., et al. Trading robustness for maintainability: An empirical study of evolving C# programs. 2014.
[6] Cacho, N., et al. How does exception handling behavior evolve? An exploratory study in Java and C# applications. 2014.

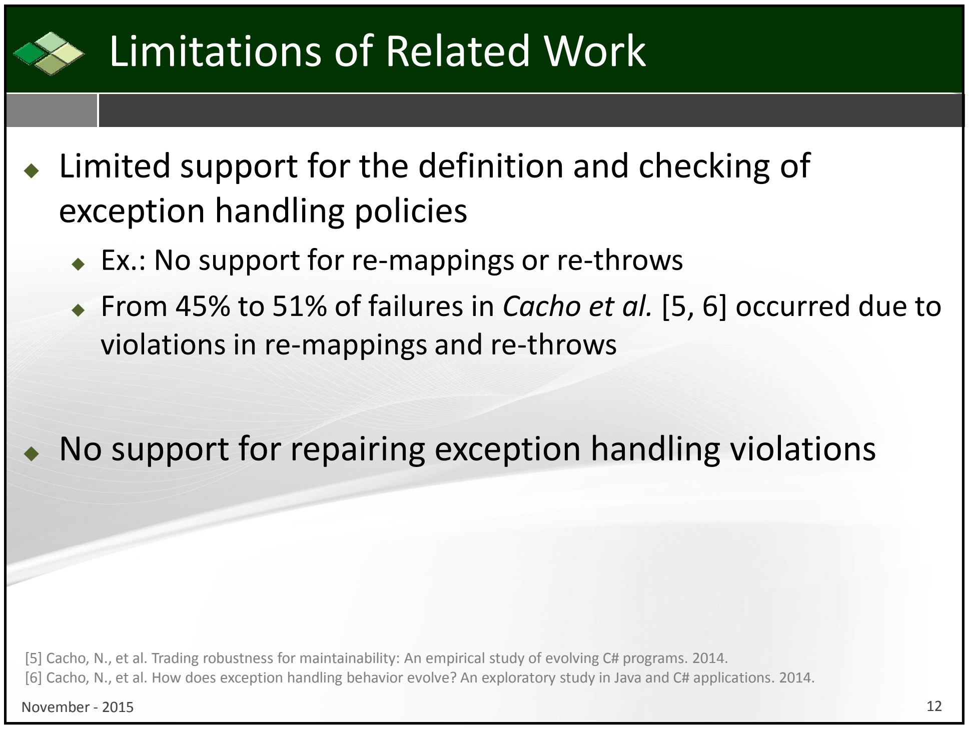
November - 2015 10



Limitations of Related Work

- Existing research on exception handling has focused on:
  - Analysis of exception handling-related failures
  - Support for exception handling comprehension
  - Support for specification and verification of exception handling properties
  - Support for exception handlers implementation

November - 2015 11

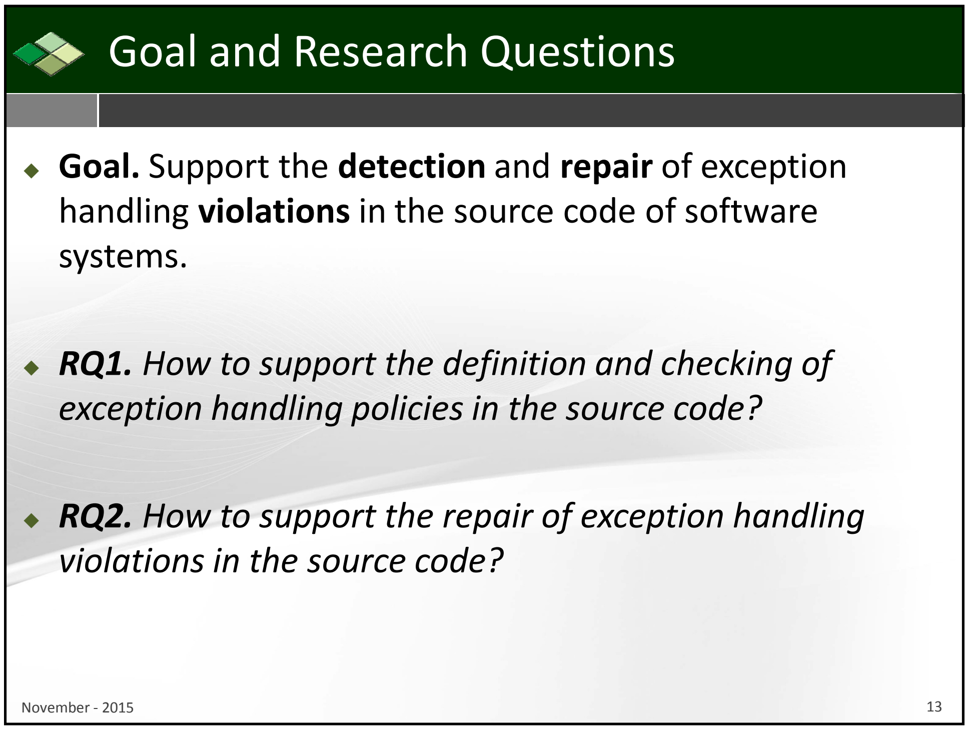


Limitations of Related Work

- Limited support for the definition and checking of exception handling policies
  - Ex.: No support for re-mappings or re-throws
  - From 45% to 51% of failures in *Cacho et al.* [5, 6] occurred due to violations in re-mappings and re-throws
- No support for repairing exception handling violations

[5] Cacho, N., et al. Trading robustness for maintainability: An empirical study of evolving C# programs. 2014.
[6] Cacho, N., et al. How does exception handling behavior evolve? An exploratory study in Java and C# applications. 2014.

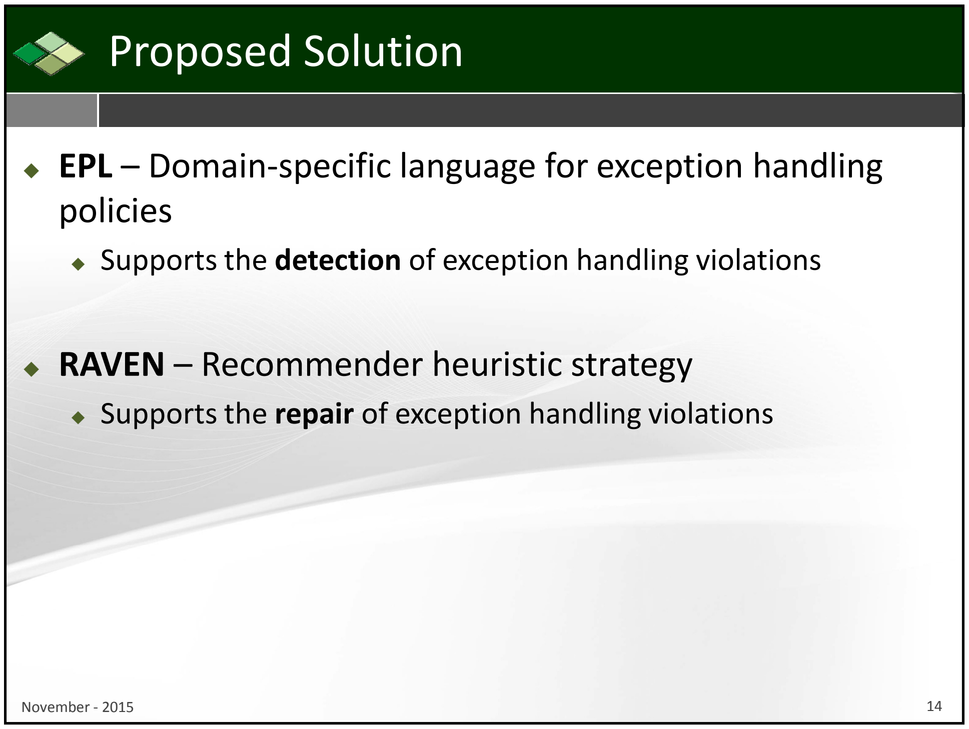
November - 2015 12



Goal and Research Questions

- **Goal.** Support the **detection** and **repair** of exception handling **violations** in the source code of software systems.
- ***RQ1.*** *How to support the definition and checking of exception handling policies in the source code?*
- ***RQ2.*** *How to support the repair of exception handling violations in the source code?*

November - 2015 13



Proposed Solution

- **EPL** – Domain-specific language for exception handling policies
  - Supports the **detection** of exception handling violations
- **RAVEN** – Recommender heuristic strategy
  - Supports the **repair** of exception handling violations

November - 2015 14

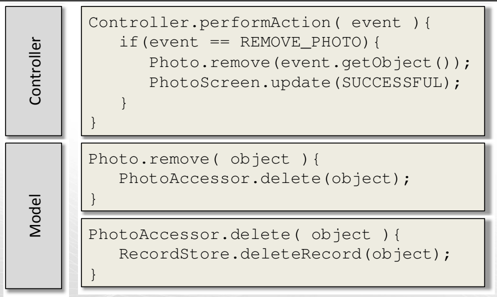
EPL - Exception handling Policies Language

- Domain-specific language to specify and verify exception handling policies
- Specify exception handling policies in terms of exception handling design rules
  - Rules expressed as **permissions** and **obligations**
    - Rules: *Only-May*, *May-Only, Cannot* and *Must*
  - Exception handling dependencies: **Handle**, **Raise**, **Propagate**, **Remap** and **Rethrow**

November - 2015 15

Exception Handling Policy Definition

EPL Syntax



```
Controller.performAction( event ){
   if(event == REMOVE_PHOTO){
      Photo.remove(event.getObject());
      PhotoScreen.update(SUCCESSFUL);
   }
}
```

```
Photo.remove( object ){
   PhotoAccessor.delete(object);
}
```

```
PhotoAccessor.delete( object ){
   RecordStore.deleteRecord(object);
}
```

```
// Compartment definition
define Model.*.* as compartment MODEL;
define Controller.*.* as compartment CONTROL;
// Rules definition
MODEL cannot handle RecordStoreException;
MODEL must remap from RecordStoreException to
PersistenceException;
CONTROLLER must handle PersistenceException;
```

16

EPL Evaluation

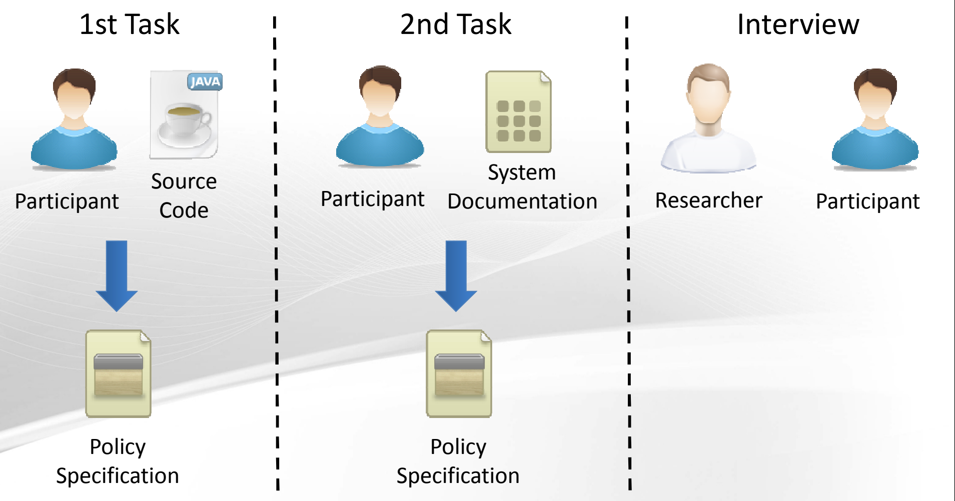
***RQ1.** How to support the **definition** and **checking** of exception handling policies in the source code?*

- **Definition** of exception handling policies
  - User-centric study
    - 10 developers-participants
- **Checking** of exception handling policies
  - Case study
    - 3 target systems

November - 2015 17

User-Centric Study

EPL Evaluation



November - 2015 18

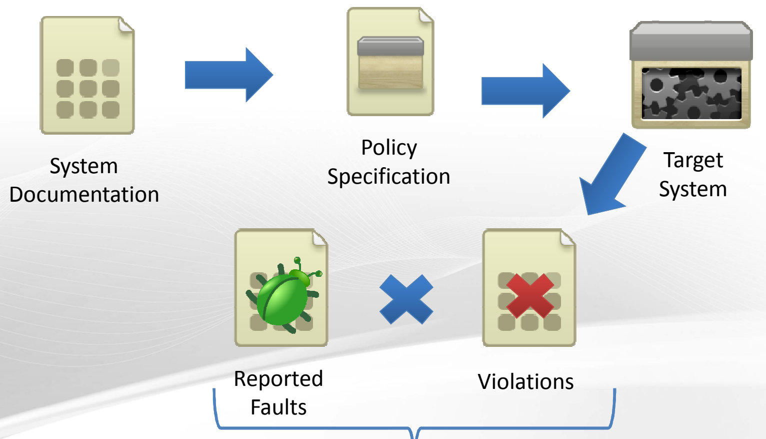
User-Centric Study – Results

EPL Evaluation

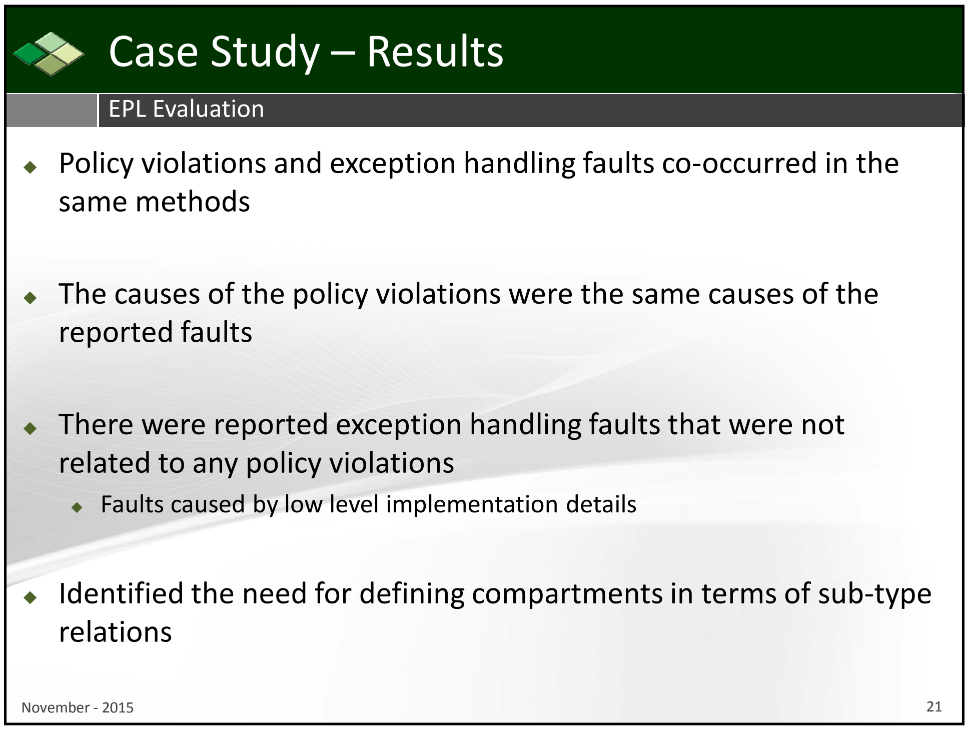
- Identified 6 factors that might influence the acceptance of EPL
  - Perceived usefulness, Expressiveness, Usability, Impact on Performance and Productivity, Learnability, Comprehensibility
- Identified characteristic that hindered the definition of exception handling policies
  - Lack of rules expressing "Prohibition"
    - New rule: *Cannot*

Case Study

EPL Evaluation



Co-occur in the same method?
Are related?

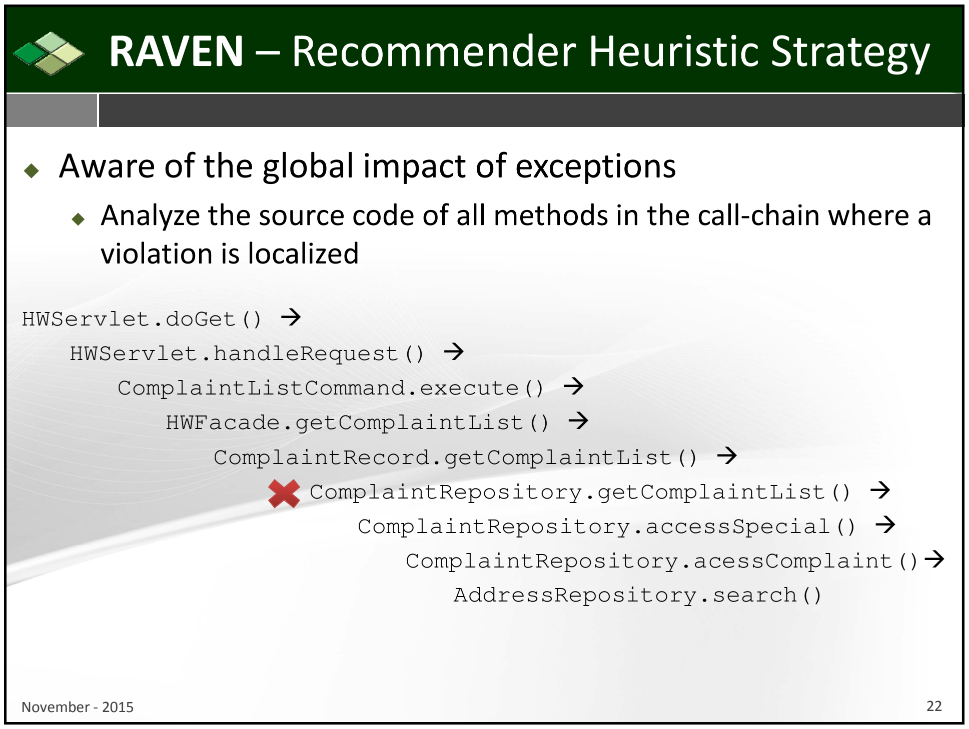


Case Study – Results

EPL Evaluation

- Policy violations and exception handling faults co-occurred in the same methods
- The causes of the policy violations were the same causes of the reported faults
- There were reported exception handling faults that were not related to any policy violations
  - Faults caused by low level implementation details
- Identified the need for defining compartments in terms of sub-type relations

November - 2015 21

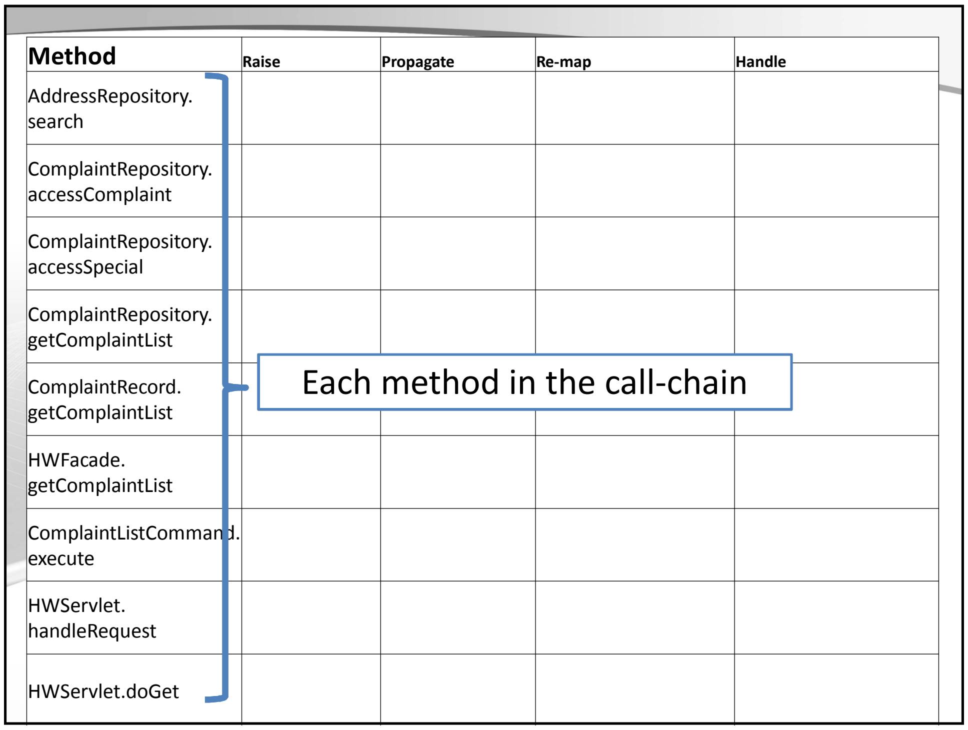


RAVEN – Recommender Heuristic Strategy

- Aware of the global impact of exceptions
  - Analyze the source code of all methods in the call-chain where a violation is localized

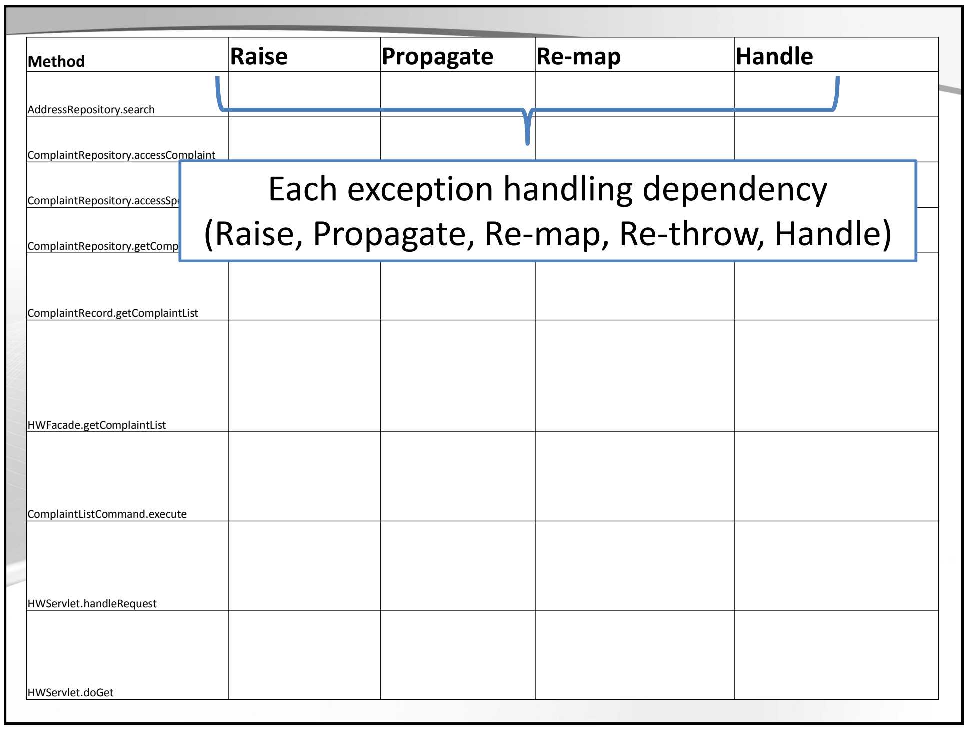
```
HWServlet.doGet() →
    HWServlet.handleRequest() →
        ComplaintListCommand.execute() →
            HWFacade.getComplaintList() →
                ComplaintRecord.getComplaintList() →
                  ✖ ComplaintRepository.getComplaintList() →
                        ComplaintRepository.accessSpecial() →
                            ComplaintRepository.acessComplaint()→
                                AddressRepository.search()
```

November - 2015 22



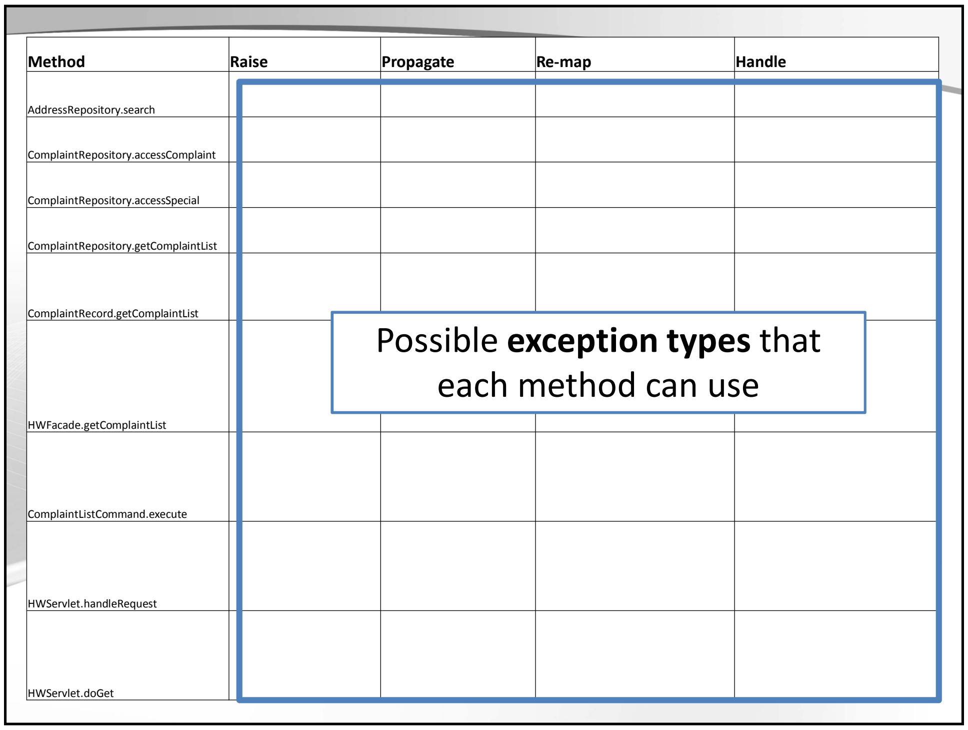
Method	Raise	Propagate	Re-map	Handle
AddressRepository. search				
ComplaintRepository. accessComplaint				
ComplaintRepository. accessSpecial				
ComplaintRepository. getComplaintList				
ComplaintRecord. getComplaintList				
HWFacade. getComplaintList				
ComplaintListCommand. execute				
HWServlet. handleRequest				
HWServlet.doGet				

Each method in the call-chain



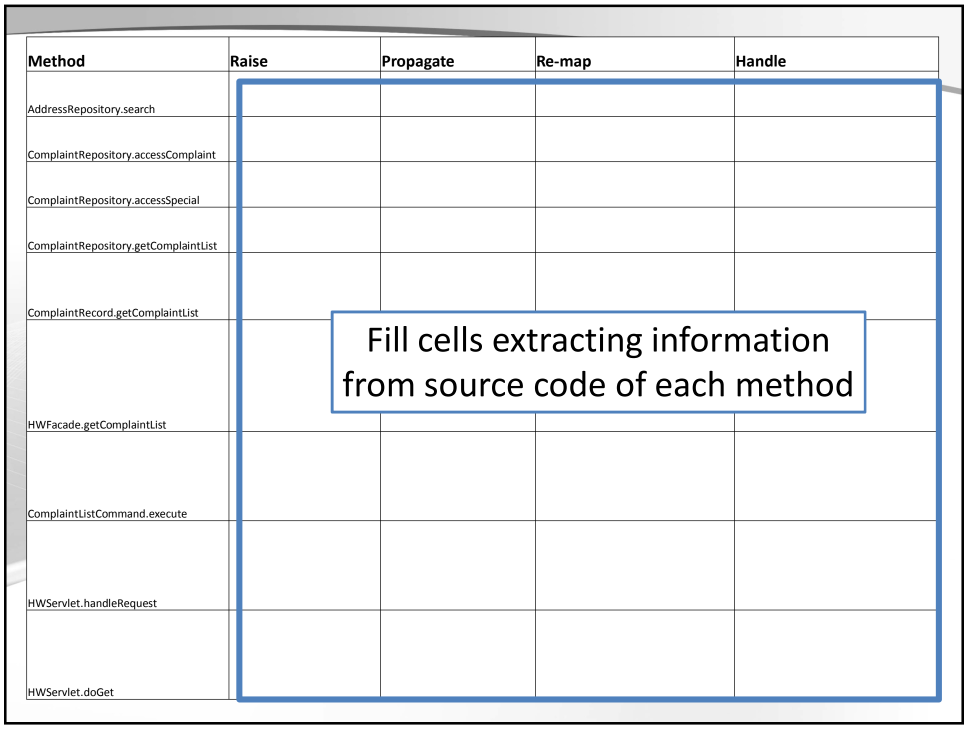
Method	Raise	Propagate	Re-map	Handle
AddressRepository.search				
ComplaintRepository.accessComplaint				
ComplaintRepository.accessSp				
ComplaintRepository.getComp				
ComplaintRecord.getComplaintList				
HWFacade.getComplaintList				
ComplaintListCommand.execute				
HWServlet.handleRequest				
HWServlet.doGet				

Each exception handling dependency
(Raise, Propagate, Re-map, Re-throw, Handle)



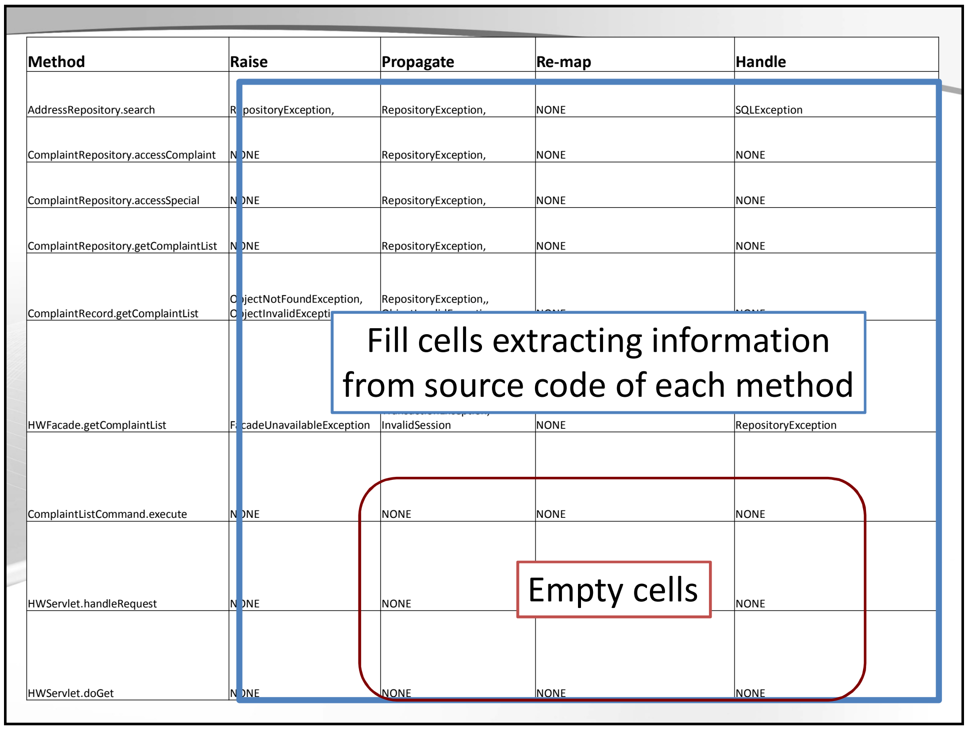
Method	Raise	Propagate	Re-map	Handle
AddressRepository.search				
ComplaintRepository.accessComplaint				
ComplaintRepository.accessSpecial				
ComplaintRepository.getComplaintList				
ComplaintRecord.getComplaintList				
HWFacade.getComplaintList				
ComplaintListCommand.execute				
HWServlet.handleRequest				
HWServlet.doGet				

Possible **exception types** that each method can use



Method	Raise	Propagate	Re-map	Handle
AddressRepository.search				
ComplaintRepository.accessComplaint				
ComplaintRepository.accessSpecial				
ComplaintRepository.getComplaintList				
ComplaintRecord.getComplaintList				
HWFacade.getComplaintList				
ComplaintListCommand.execute				
HWServlet.handleRequest				
HWServlet.doGet				

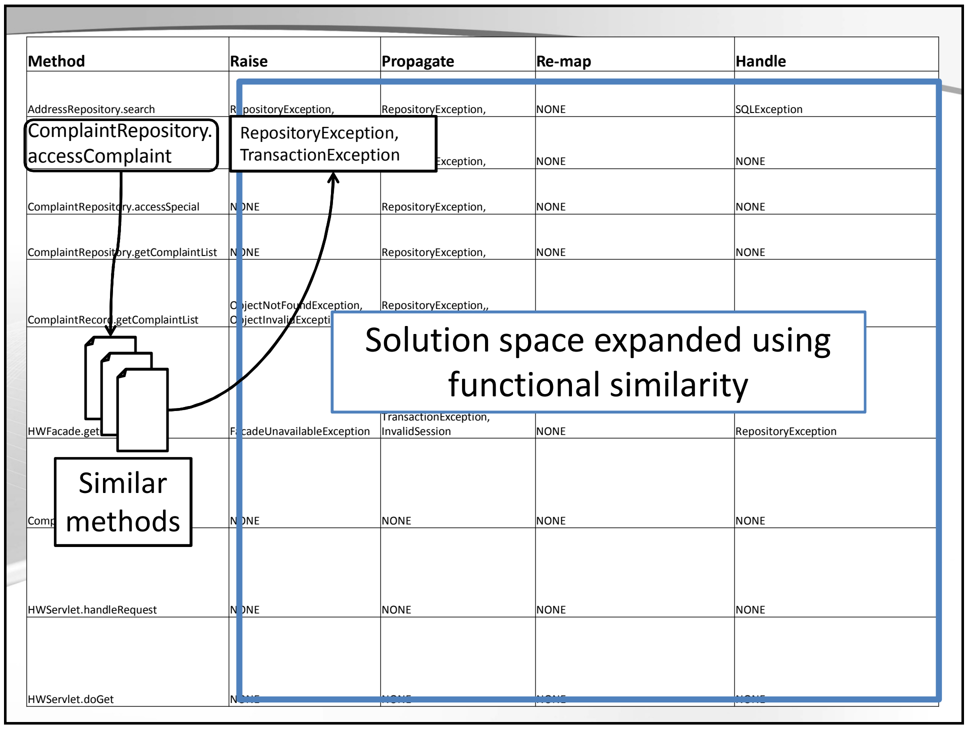
Fill cells extracting information from source code of each method



Method	Raise	Propagate	Re-map	Handle
AddressRepository.search	RepositoryException,	RepositoryException,	NONE	SQLException
ComplaintRepository.accessComplaint	NONE	RepositoryException,	NONE	NONE
ComplaintRepository.accessSpecial	NONE	RepositoryException,	NONE	NONE
ComplaintRepository.getComplaintList	NONE	RepositoryException,	NONE	NONE
ComplaintRecord.getComplaintList	ObjectNotFoundException, ObjectInvalidExcepti	RepositoryException,,	NONE	NONE
HWFacade.getComplaintList	FacadeUnavailableException	InvalidSession	NONE	RepositoryException
ComplaintListCommand.execute	NONE	NONE	NONE	NONE
HWServlet.handleRequest	NONE	NONE		NONE
HWServlet.doGet	NONE	NONE	NONE	NONE

Fill cells extracting information from source code of each method

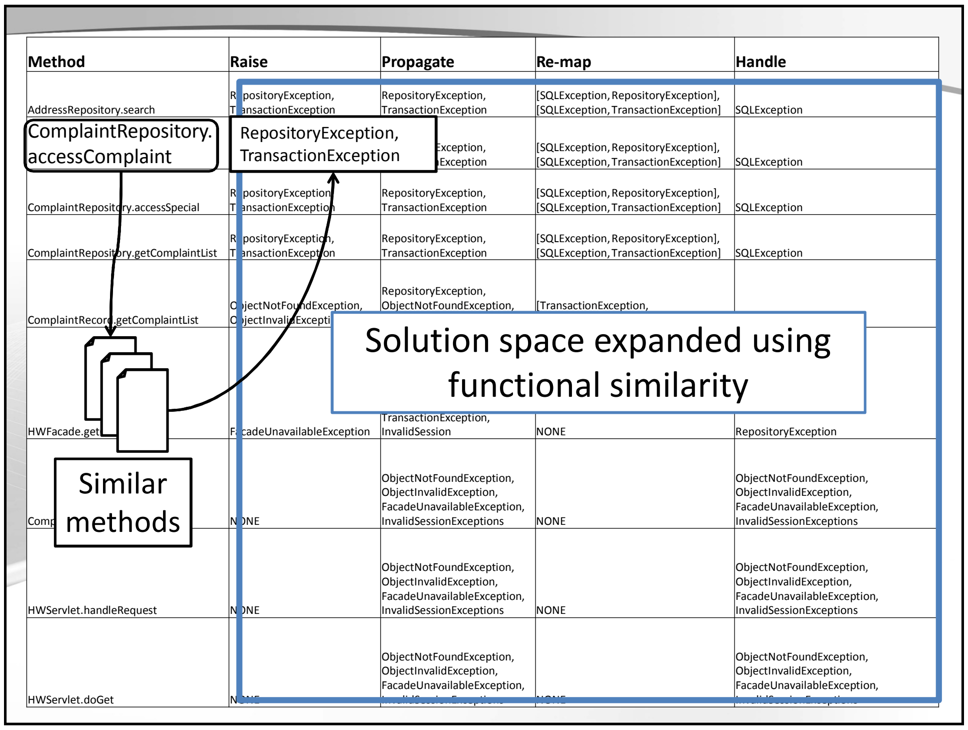
Empty cells



Method	Raise	Propagate	Re-map	Handle
AddressRepository.search	RepositoryException,	RepositoryException,	NONE	SQLException
ComplaintRepository.accessComplaint	RepositoryException, TransactionException	Exception,	NONE	NONE
ComplaintRepository.accessSpecial	NONE	RepositoryException,	NONE	NONE
ComplaintRepository.getComplaintList	NONE	RepositoryException,	NONE	NONE
ComplaintRecord.getComplaintList	ObjectNotFoundException, ObjectInvalidExcepti	RepositoryException,,		
HWFacade.get	FacadeUnavailableException	TransactionException, InvalidSession	NONE	RepositoryException
Comp	NONE	NONE	NONE	NONE
HWServlet.handleRequest	NONE	NONE	NONE	NONE
HWServlet.doGet	NONE	NONE	NONE	NONE

Similar methods

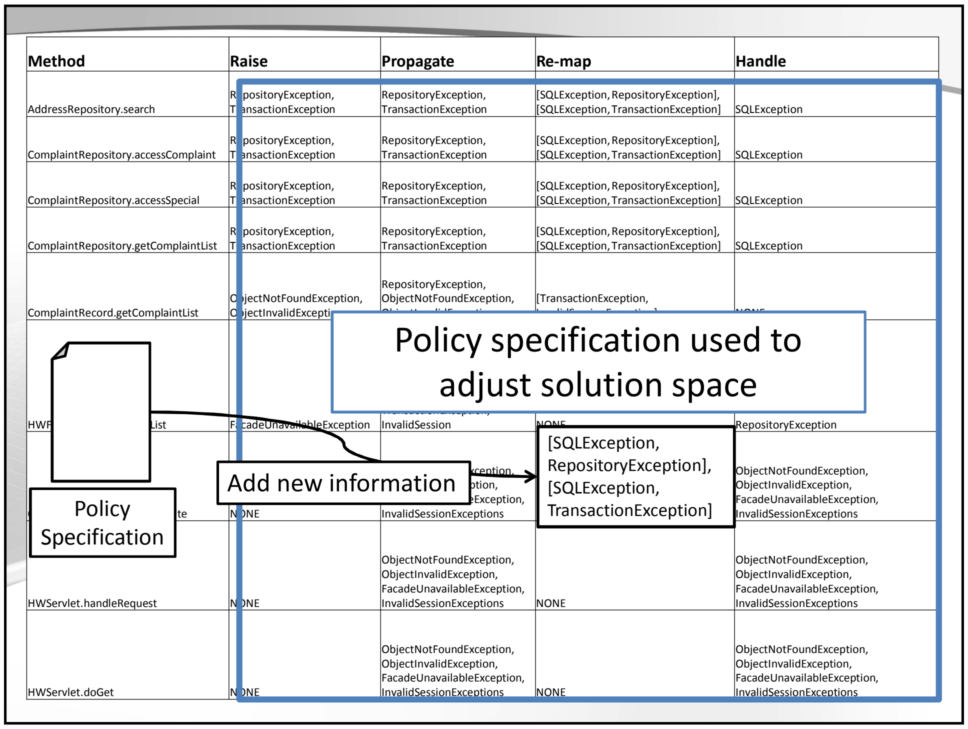
Solution space expanded using functional similarity



Method	Raise	Propagate	Re-map	Handle
AddressRepository.search	RepositoryException, TransactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository. accessComplaint	RepositoryException, TransactionException	xception, Exception	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.accessSpecial	RepositoryException TransactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.getComplaintList	RepositoryException, TransactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRecord.getComplaintList	ObjectNotFoundException, ObjectInvalidExcepti	RepositoryException, ObjectNotFoundException,	[TransactionException,	
HWFacade.get	FacadeUnavailableException	TransactionException, InvalidSession	NONE	RepositoryException
Comp	NONE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions	NONE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions
HWServlet.handleRequest	NONE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions	NONE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions
HWServlet.doGet	NONE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException,		ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException,

Solution space expanded using functional similarity

Similar methods

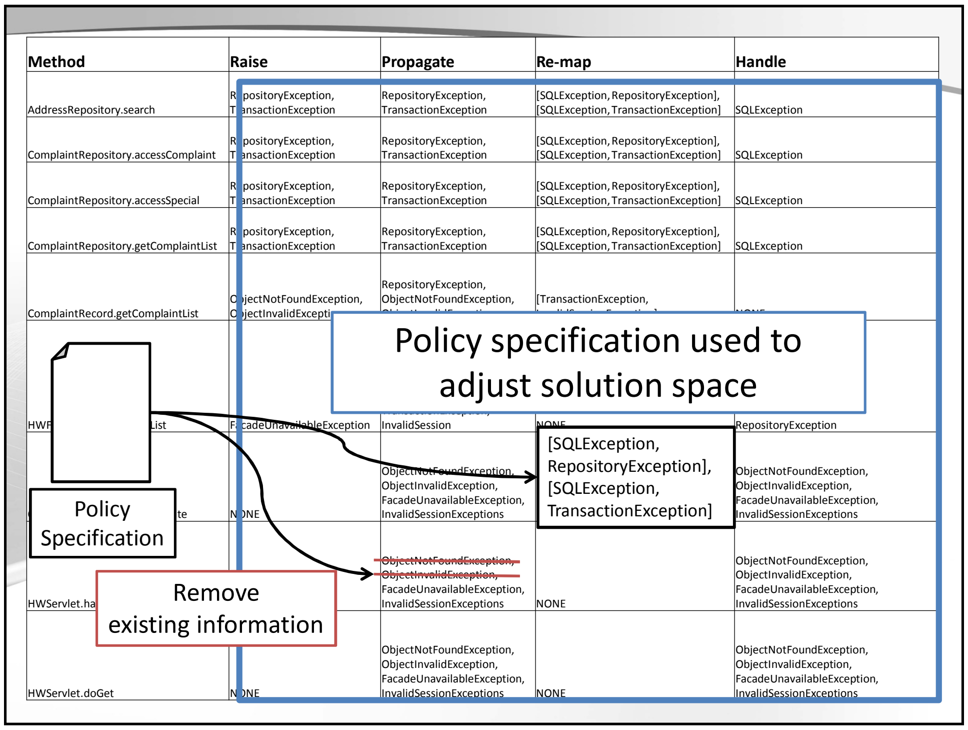


Method	Raise	Propagate	Re-map	Handle
AddressRepository.search	RepositoryException, TransactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.accessComplaint	RepositoryException, TransactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.accessSpecial	RepositoryException, TransactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.getComplaintList	RepositoryException, TransactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRecord.getComplaintList	ObjectNotFoundException, ObjectInvalidExcepti	RepositoryException, ObjectNotFoundException,	[TransactionException,	NONE
HWF List	FacadeUnavailableException	InvalidSession	NONE	RepositoryException
te	NONE	xception, ption, Exception, InvalidSessionExceptions	[SQLException, RepositoryException], [SQLException, TransactionException]	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions
HWServlet.handleRequest	NONE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions	NONE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions
HWServlet.doGet	NONE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions	NONE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions

Policy specification used to adjust solution space

Add new information

Policy Specification

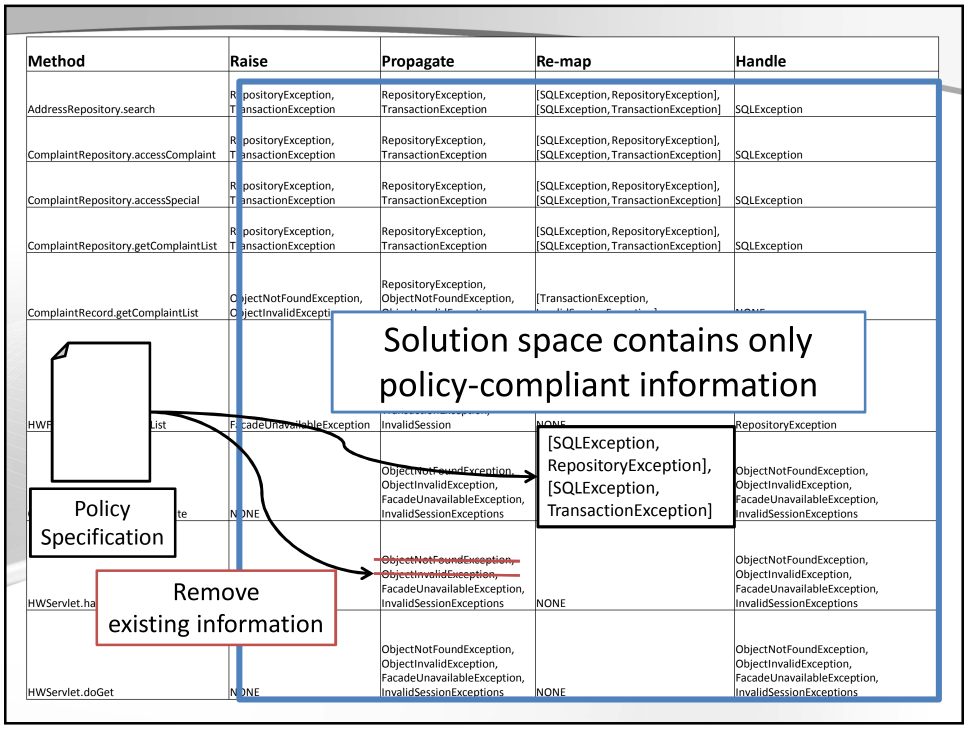


Policy specification used to adjust solution space

Method	Raise	Propagate	Re-map	Handle
AddressRepository.search	R positoryException, T ansactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.accessComplaint	R positoryException, T ansactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.accessSpecial	R positoryException, T ansactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.getComplaintList	R positoryException, T ansactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRecord.getComplaintList	O jectNotFoundException, O jectInvalidExcepti	RepositoryException, ObjectNotFoundException,	[TransactionException,	
HWF List	F cadeUnavailableException	InvalidSession		RepositoryException
te	N NE	~~ObjectNotFoundException,~~ ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions	[SQLException, RepositoryException], [SQLException, TransactionException]	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions
HWServlet.ha		~~ObjectNotFoundException, ObjectInvalidException,~~ FacadeUnavailableException, InvalidSessionExceptions	NONE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions
HWServlet.doGet	N NE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions	NONE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions

Policy Specification

Remove existing information

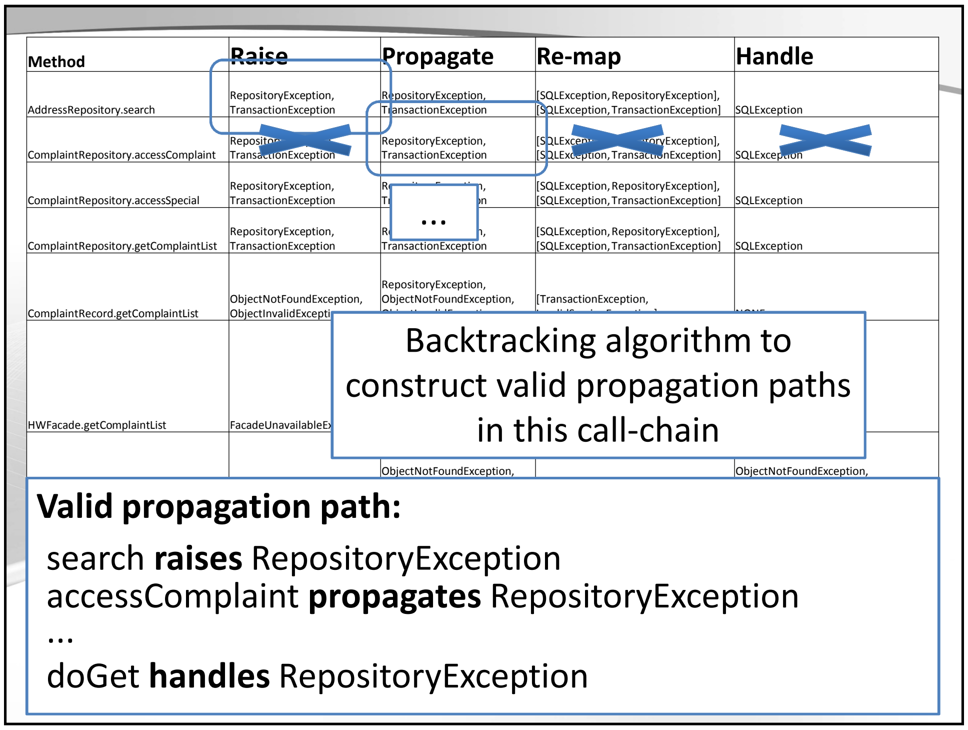


Solution space contains only policy-compliant information

Method	Raise	Propagate	Re-map	Handle
AddressRepository.search	R positoryException, T ansactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.accessComplaint	R positoryException, T ansactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.accessSpecial	R positoryException, T ansactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.getComplaintList	R positoryException, T ansactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRecord.getComplaintList	O jectNotFoundException, O jectInvalidExcepti	RepositoryException, ObjectNotFoundException,	[TransactionException,	
HWF List	F cadeUnavailableException	InvalidSession		RepositoryException
te	N NE	~~ObjectNotFoundException,~~ ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions	[SQLException, RepositoryException], [SQLException, TransactionException]	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions
HWServlet.ha		~~ObjectNotFoundException, ObjectInvalidException,~~ FacadeUnavailableException, InvalidSessionExceptions	NONE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions
HWServlet.doGet	N NE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions	NONE	ObjectNotFoundException, ObjectInvalidException, FacadeUnavailableException, InvalidSessionExceptions

Policy Specification

Remove existing information

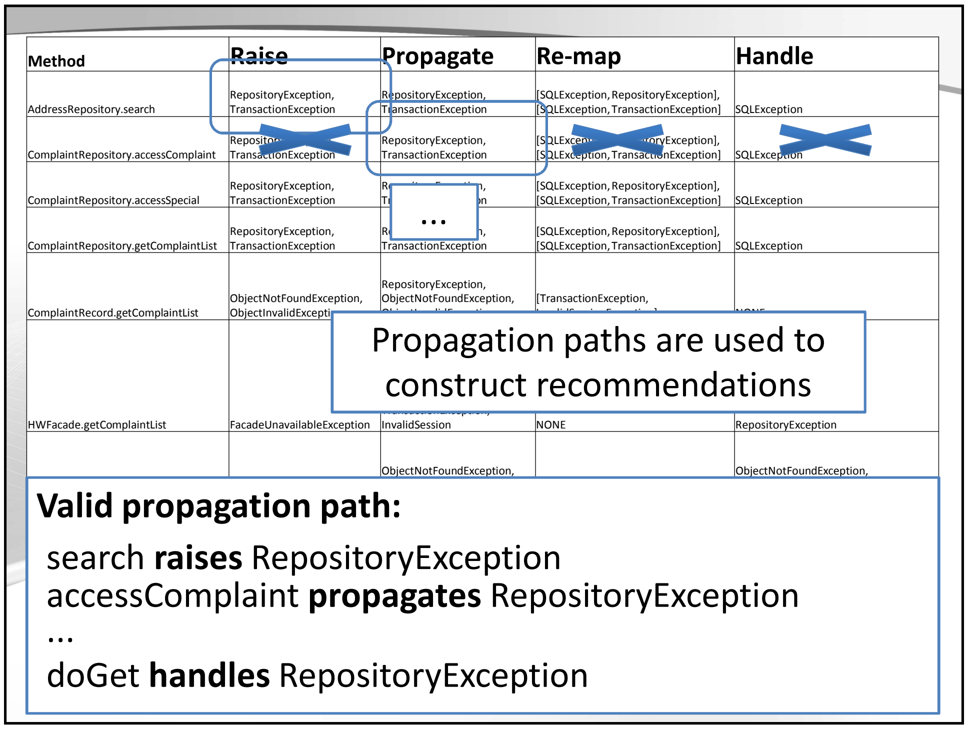


Method	Raise	Propagate	Re-map	Handle
AddressRepository.search	RepositoryException, TransactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.accessComplaint	Reposito TransactionException	RepositoryException, TransactionException	[SQLExcep oryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.accessSpecial	RepositoryException, TransactionException	...	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.getComplaintList	RepositoryException, TransactionException	TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRecord.getComplaintList	ObjectNotFoundException, ObjectInvalidExcepti	RepositoryException, ObjectNotFoundException,	[TransactionException,	
HWFacade.getComplaintList	FacadeUnavailableEx			
		ObjectNotFoundException,		ObjectNotFoundException,

Backtracking algorithm to construct valid propagation paths in this call-chain

Valid propagation path:

search **raises** RepositoryException
accessComplaint **propagates** RepositoryException
…
doGet **handles** RepositoryException

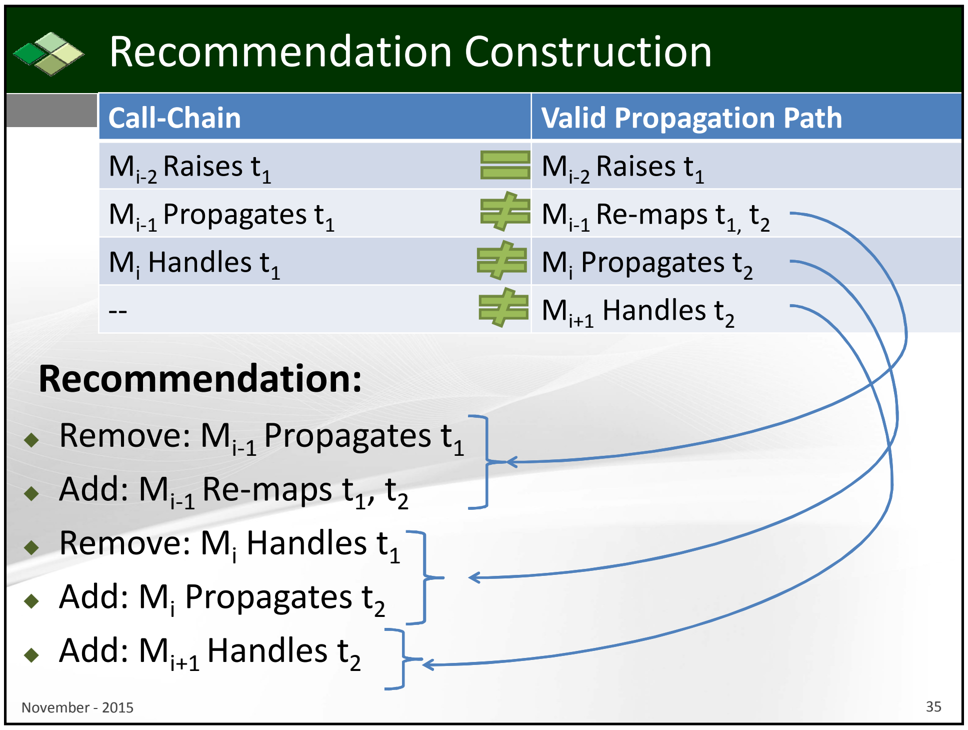


Method	Raise	Propagate	Re-map	Handle
AddressRepository.search	RepositoryException, TransactionException	RepositoryException, TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.accessComplaint	Reposito TransactionException	RepositoryException, TransactionException	[SQLExcep oryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.accessSpecial	RepositoryException, TransactionException	...	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRepository.getComplaintList	RepositoryException, TransactionException	TransactionException	[SQLException, RepositoryException], [SQLException, TransactionException]	SQLException
ComplaintRecord.getComplaintList	ObjectNotFoundException, ObjectInvalidExcepti	RepositoryException, ObjectNotFoundException,	[TransactionException,	
HWFacade.getComplaintList	FacadeUnavailableException	InvalidSession	NONE	RepositoryException
		ObjectNotFoundException,		ObjectNotFoundException,

Propagation paths are used to construct recommendations

Valid propagation path:

search **raises** RepositoryException
accessComplaint **propagates** RepositoryException
…
doGet **handles** RepositoryException



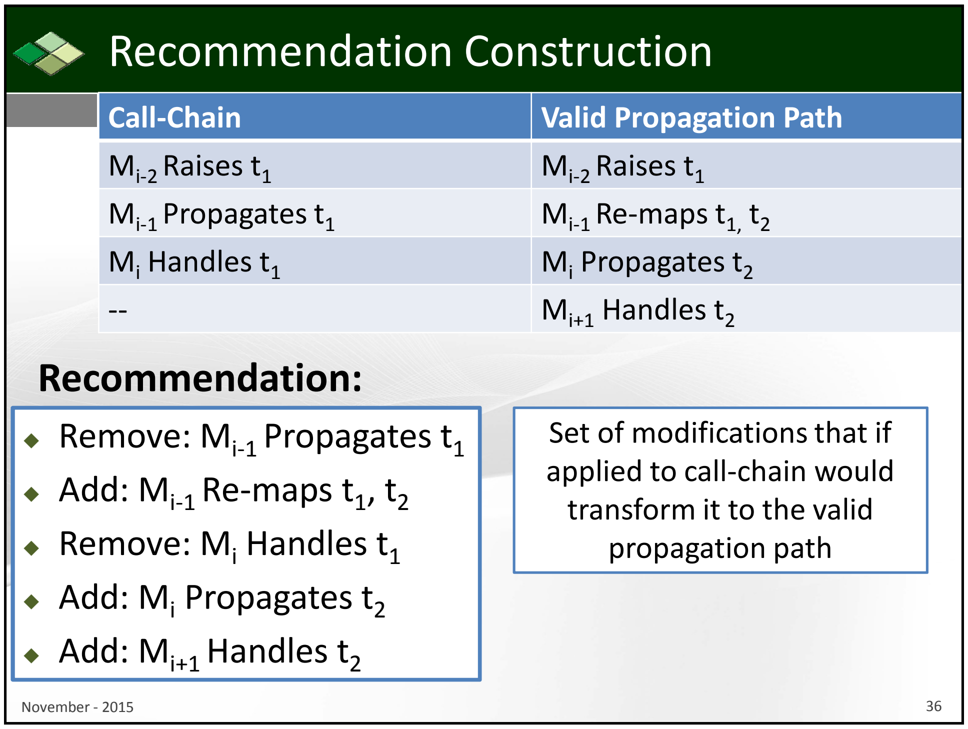
Recommendation Construction

Call-Chain		Valid Propagation Path
M_{i-2} Raises t_1	=	M_{i-2} Raises t_1
M_{i-1} Propagates t_1	≠	M_{i-1} Re-maps t_1, t_2
M_i Handles t_1	≠	M_i Propagates t_2
--	≠	M_{i+1} Handles t_2

Recommendation:

- Remove: M_{i-1} Propagates t_1
- Add: M_{i-1} Re-maps t_1, t_2
- Remove: M_i Handles t_1
- Add: M_i Propagates t_2
- Add: M_{i+1} Handles t_2

November - 2015 35



Recommendation Construction

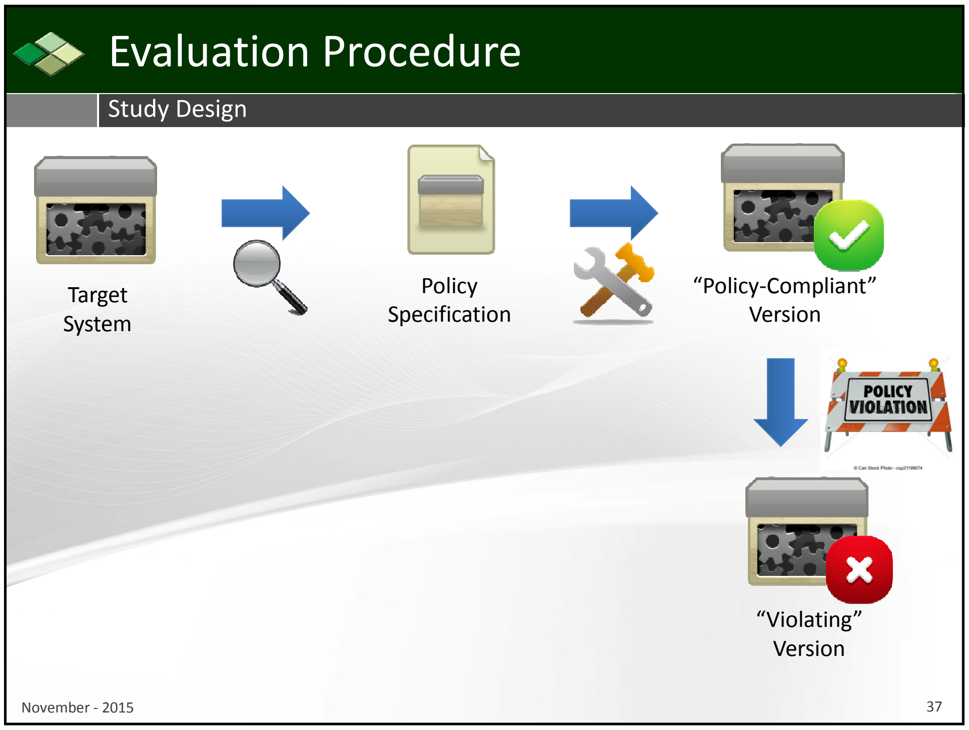
Call-Chain	Valid Propagation Path
M_{i-2} Raises t_1	M_{i-2} Raises t_1
M_{i-1} Propagates t_1	M_{i-1} Re-maps t_1, t_2
M_i Handles t_1	M_i Propagates t_2
--	M_{i+1} Handles t_2

Recommendation:

- Remove: M_{i-1} Propagates t_1
- Add: M_{i-1} Re-maps t_1, t_2
- Remove: M_i Handles t_1
- Add: M_i Propagates t_2
- Add: M_{i+1} Handles t_2

Set of modifications that if applied to call-chain would transform it to the valid propagation path

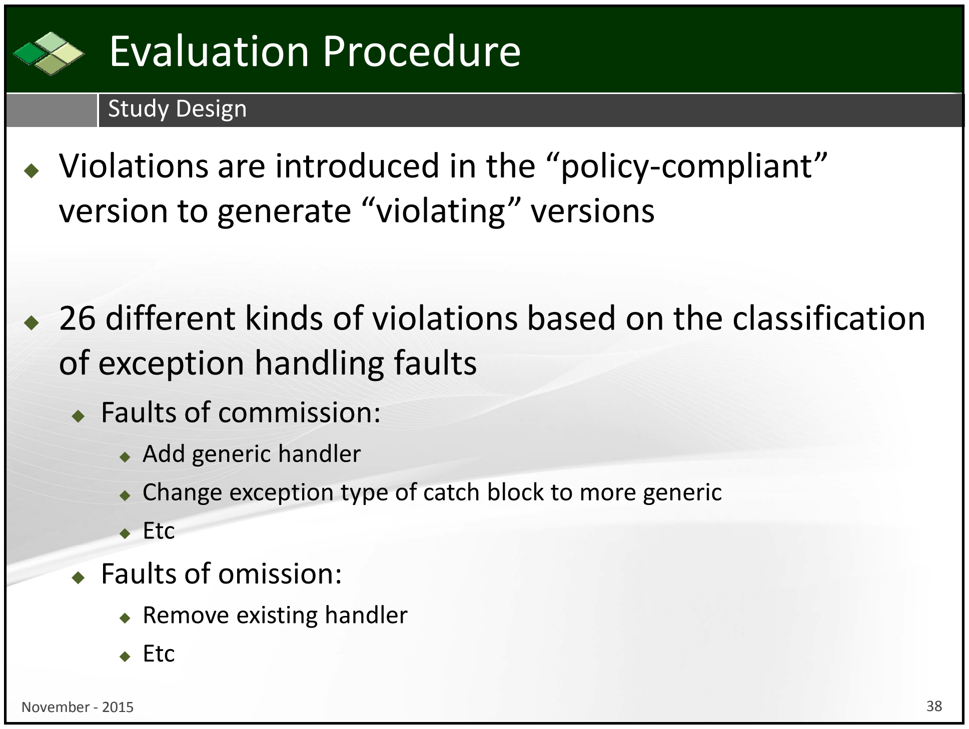
November - 2015 36



Evaluation Procedure

Study Design

November - 2015 37

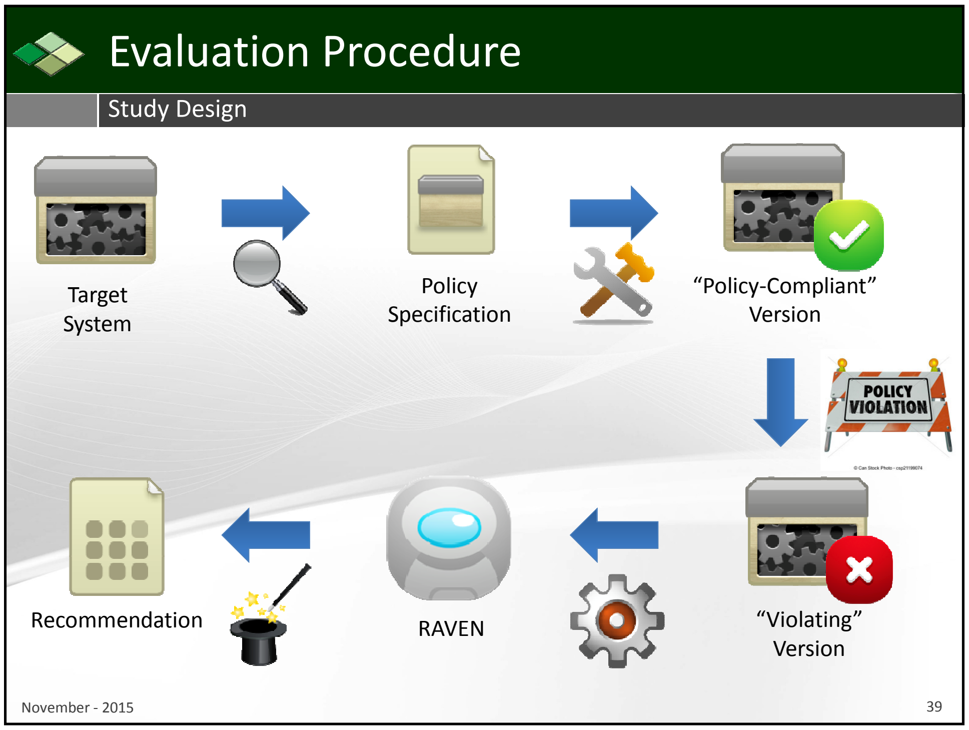


Evaluation Procedure

Study Design

- Violations are introduced in the "policy-compliant" version to generate "violating" versions
- 26 different kinds of violations based on the classification of exception handling faults
  - Faults of commission:
    - Add generic handler
    - Change exception type of catch block to more generic
    - Etc
  - Faults of omission:
    - Remove existing handler
    - Etc

November - 2015 38

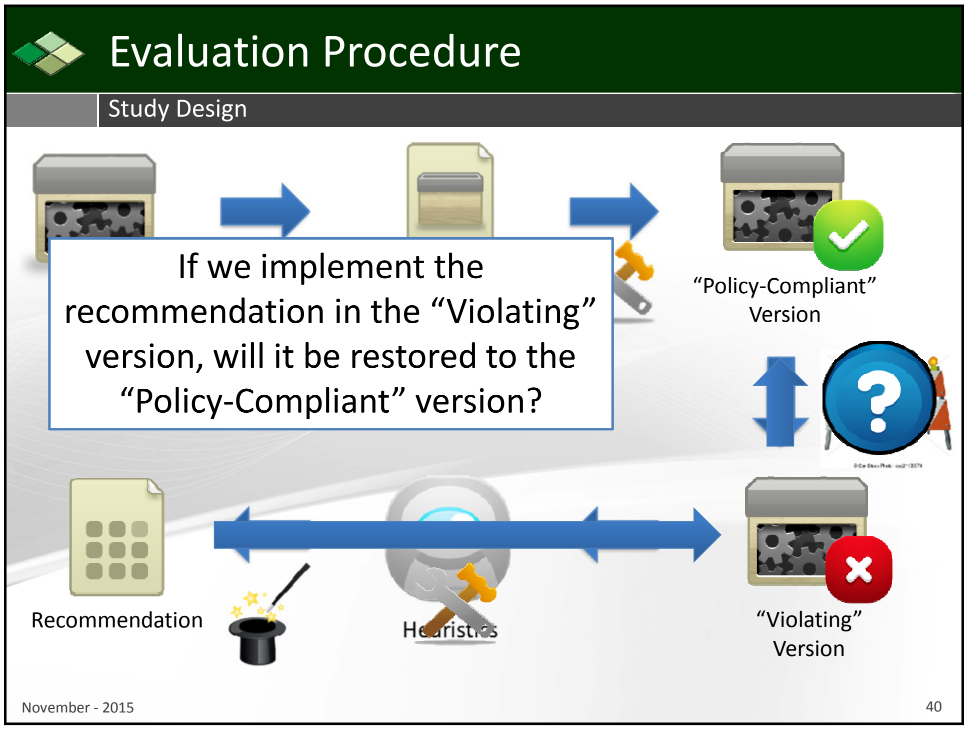


Evaluation Procedure

Study Design

November - 2015

39



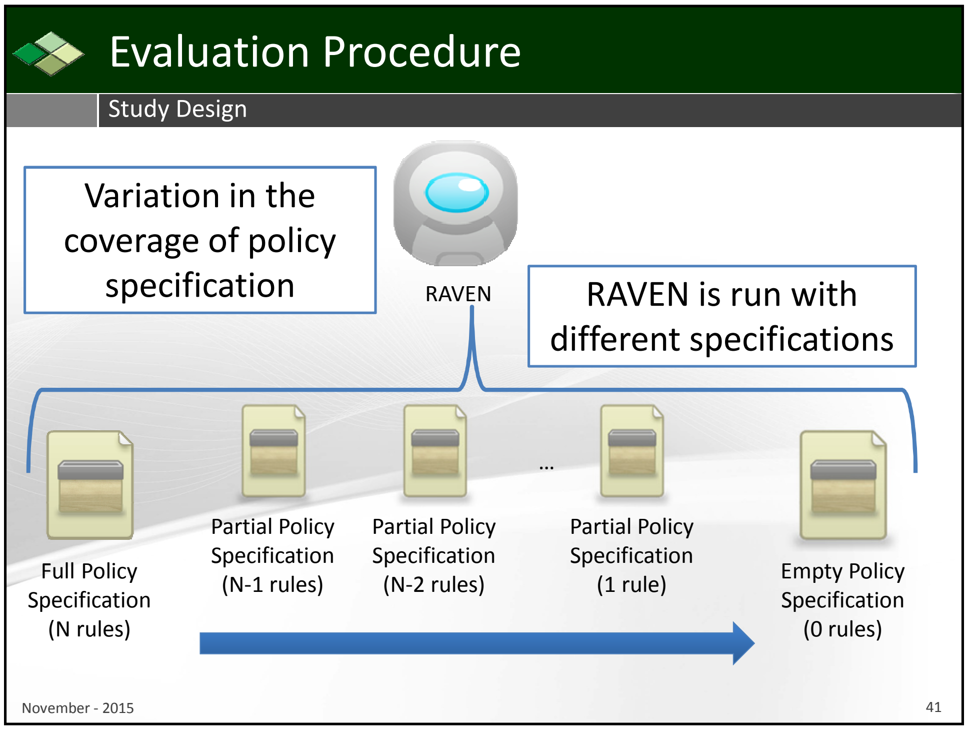
Evaluation Procedure

Study Design

If we implement the recommendation in the "Violating" version, will it be restored to the "Policy-Compliant" version?

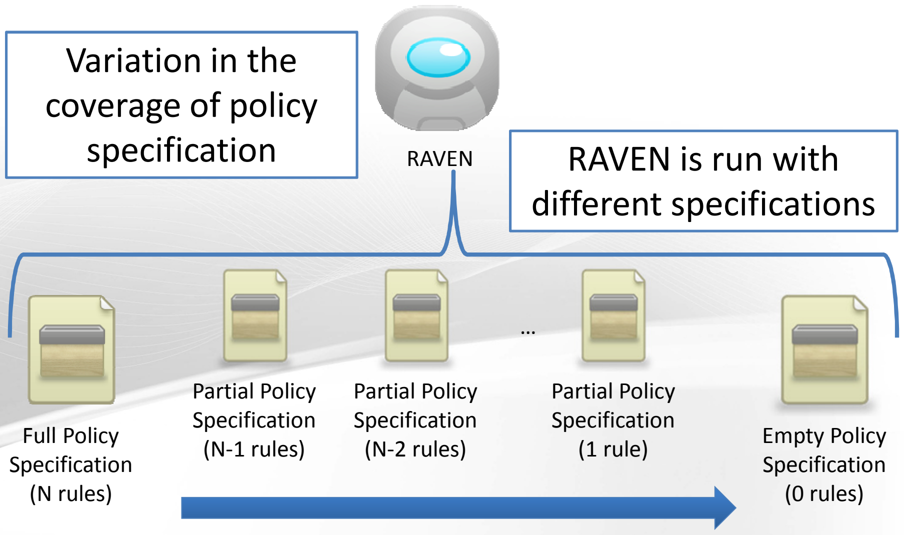
November - 2015

40

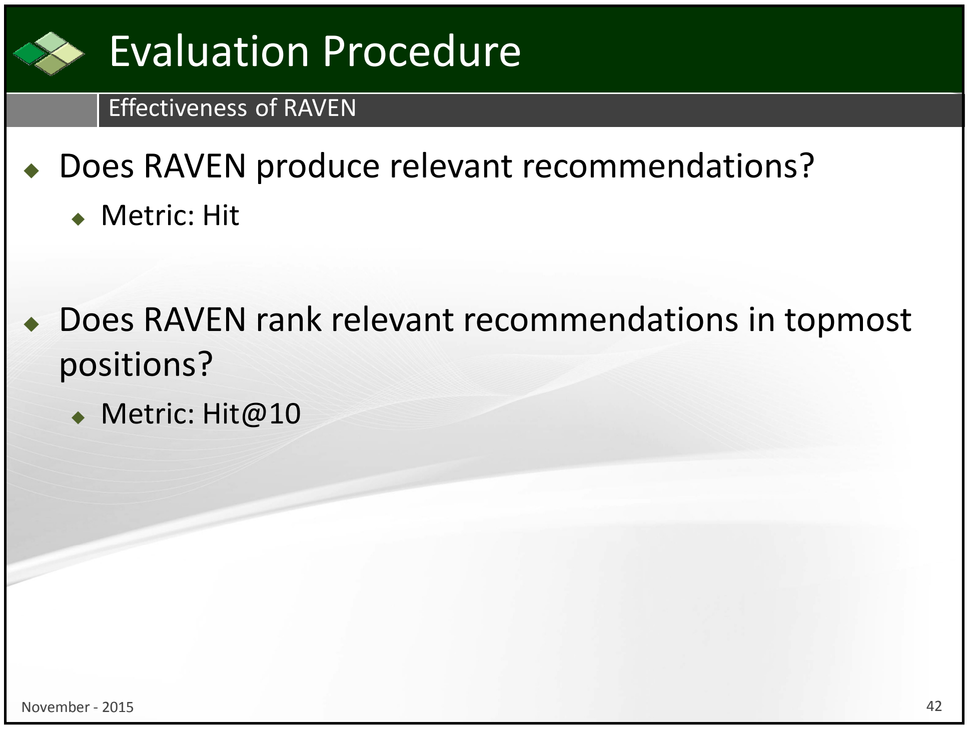


Evaluation Procedure

Study Design


Variation in the coverage of policy specification
RAVEN
RAVEN is run with different specifications
Full Policy Specification (N rules)
Partial Policy Specification (N-1 rules)
Partial Policy Specification (N-2 rules)
…
Partial Policy Specification (1 rule)
Empty Policy Specification (0 rules)

November - 2015 41



Evaluation Procedure

Effectiveness of RAVEN

- Does RAVEN produce relevant recommendations?
  - Metric: Hit
- Does RAVEN rank relevant recommendations in topmost positions?
  - Metric: Hit@10

November - 2015 42

Results: Hit and Hit@10

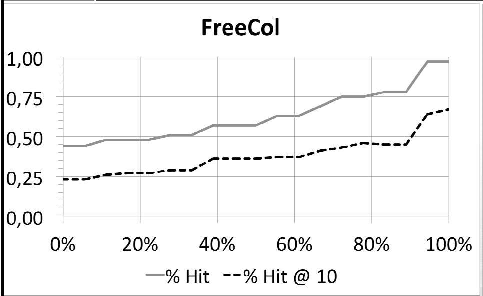
Effectiveness of RAVEN

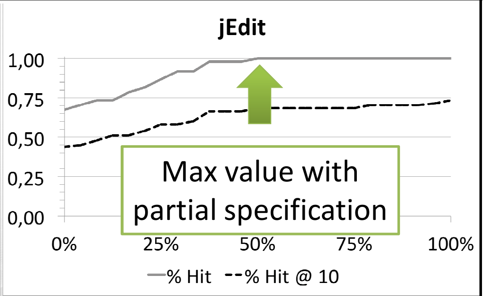
- Does RAVEN produce relevant recommendations?
  - Without Policy: Hit = 0.69
  - With Policy: Hit = 0.97
- Does RAVEN rank relevant recommendations in topmost positions?
  - Without Policy: Hit@10 = 0.43
  - With Policy: Hit@10 = 0.81

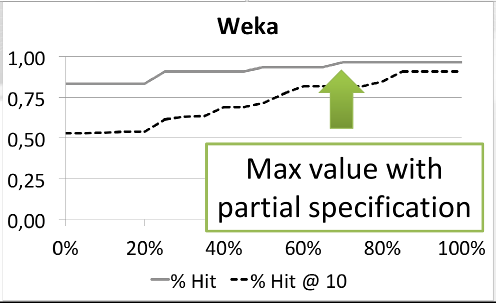
November - 2015 43

Results: Hit

Does RAVEN produce relevant recommendations?



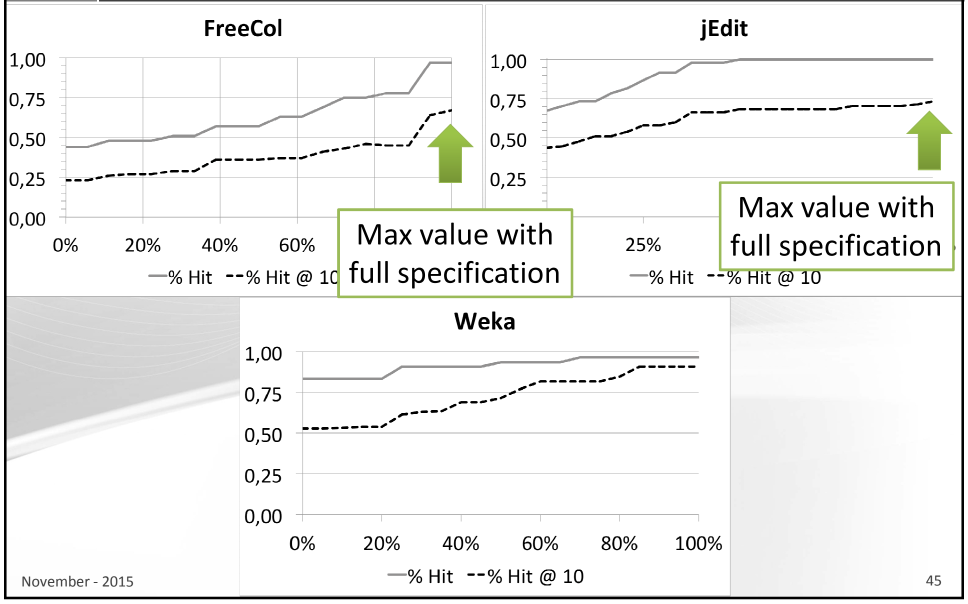




November - 2015 44

Results: Hit@10

Does RAVEN rank relevant recommendations in topmost positions?



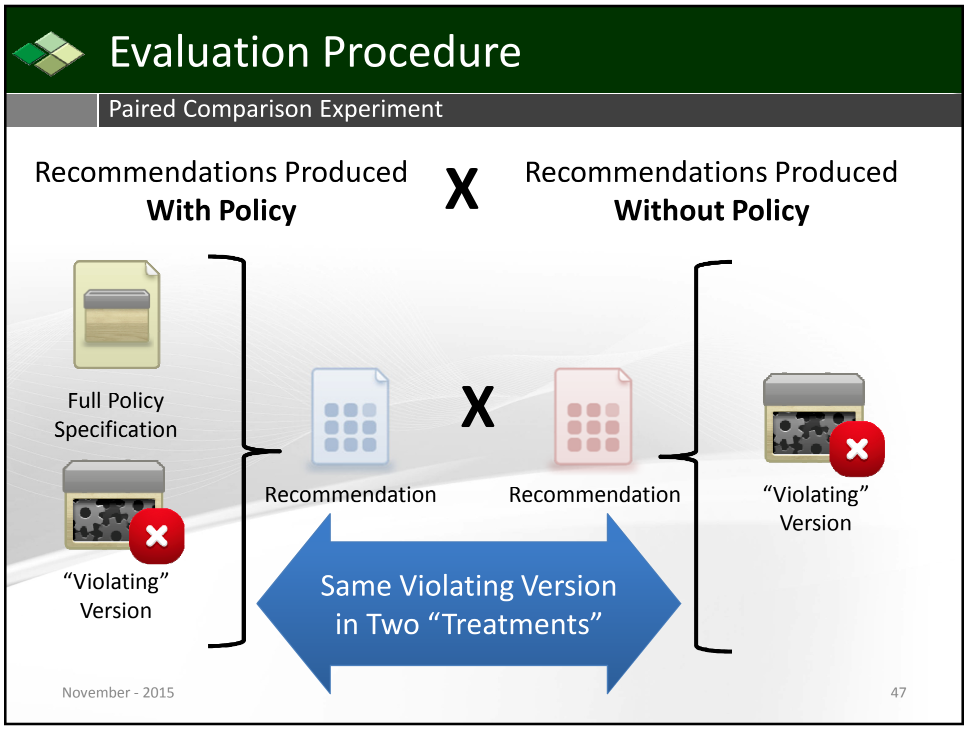
Evaluation Procedure

Effects of Using Policy Specification

- Is the effectiveness of RAVEN affected by the use of policy specifications?
  - Metric: Reciprocal Rank

$$RR(\mathbb{L}) = \frac{1}{rank(\mathbb{R})}$$

- Hypothesis
  - Null: The effectiveness of RAVEN is not affected by the use of policy specifications
  - Alternative: The effectiveness of RAVEN is affected by the use of policy specifications



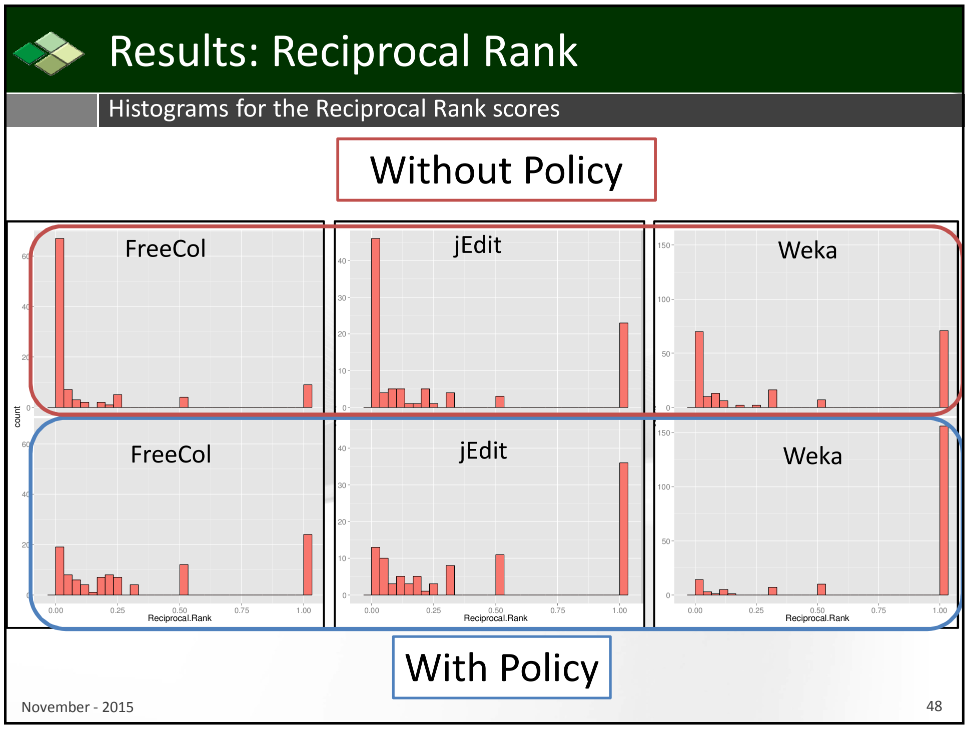
Evaluation Procedure

Paired Comparison Experiment

Recommendations Produced **With Policy** X Recommendations Produced **Without Policy**

Full Policy Specification
"Violating" Version
Recommendation
X
Recommendation
"Violating" Version
Same Violating Version in Two "Treatments"

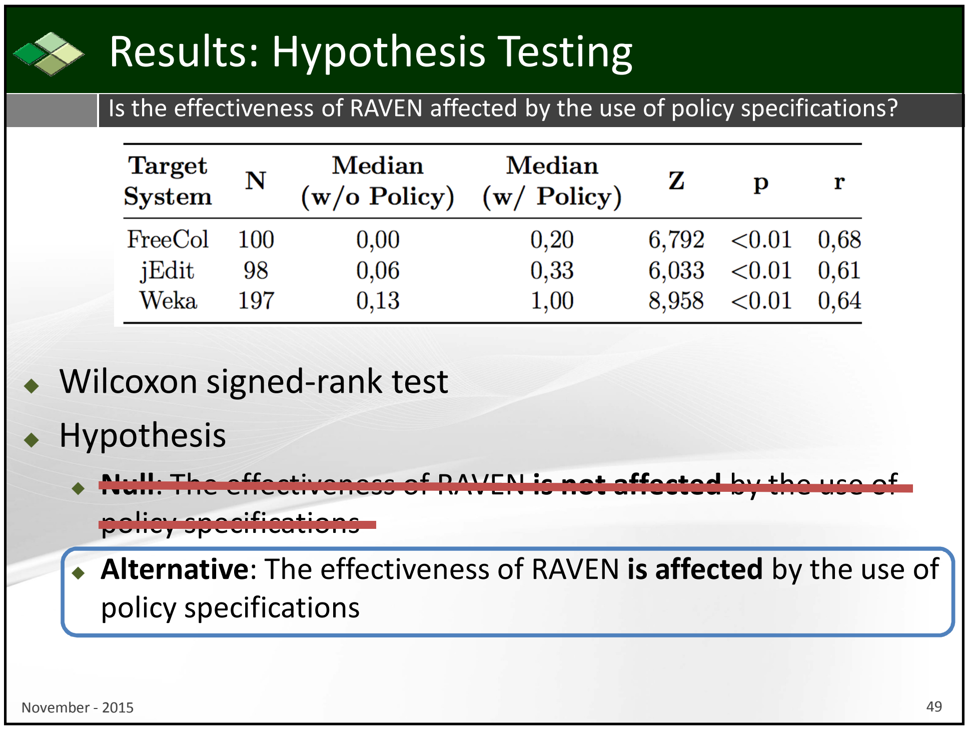
November - 2015 47



Results: Reciprocal Rank

Histograms for the Reciprocal Rank scores

November - 2015 48



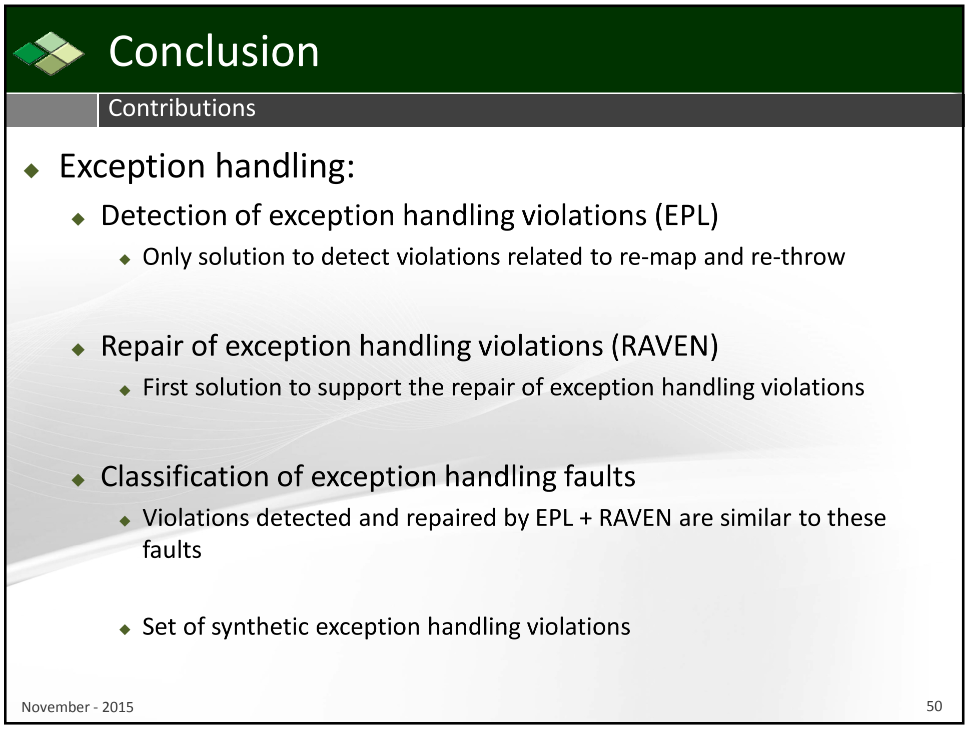
Results: Hypothesis Testing

Is the effectiveness of RAVEN affected by the use of policy specifications?

Target System	N	Median (w/o Policy)	Median (w/ Policy)	Z	p	r
FreeCol	100	0,00	0,20	6,792	<0.01	0,68
jEdit	98	0,06	0,33	6,033	<0.01	0,61
Weka	197	0,13	1,00	8,958	<0.01	0,64

- Wilcoxon signed-rank test
- Hypothesis
  - ~~**Null**: The effectiveness of RAVEN **is not affected** by the use of policy specifications~~
  - **Alternative**: The effectiveness of RAVEN **is affected** by the use of policy specifications

November - 2015 49

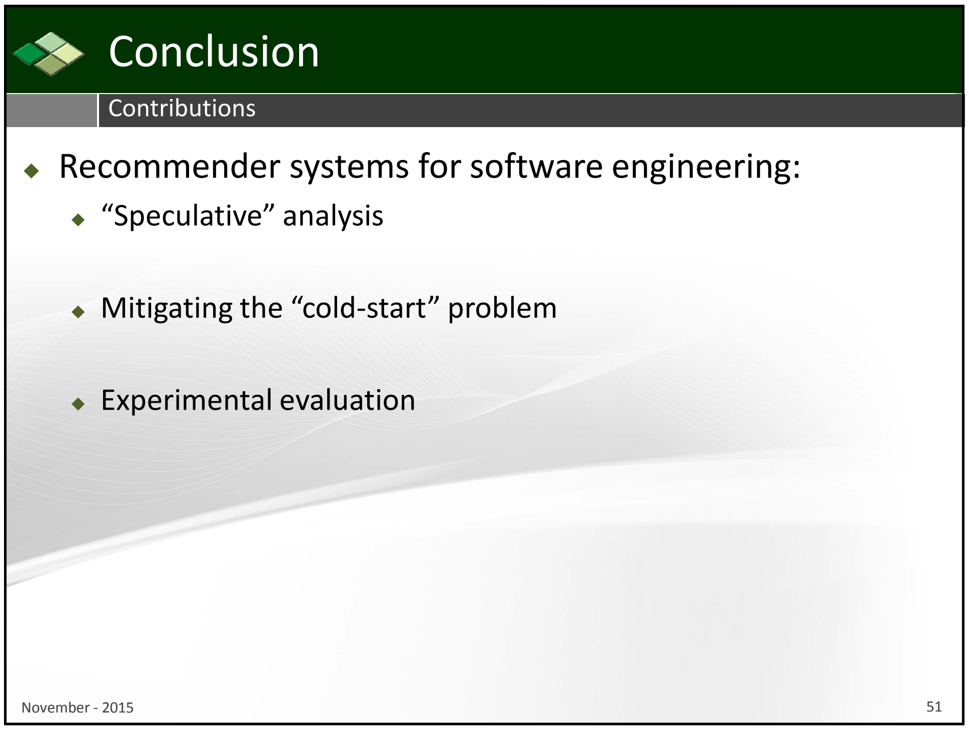


Conclusion

Contributions

- Exception handling:
  - Detection of exception handling violations (EPL)
    - Only solution to detect violations related to re-map and re-throw
  - Repair of exception handling violations (RAVEN)
    - First solution to support the repair of exception handling violations
  - Classification of exception handling faults
    - Violations detected and repaired by EPL + RAVEN are similar to these faults
    - Set of synthetic exception handling violations

November - 2015 50

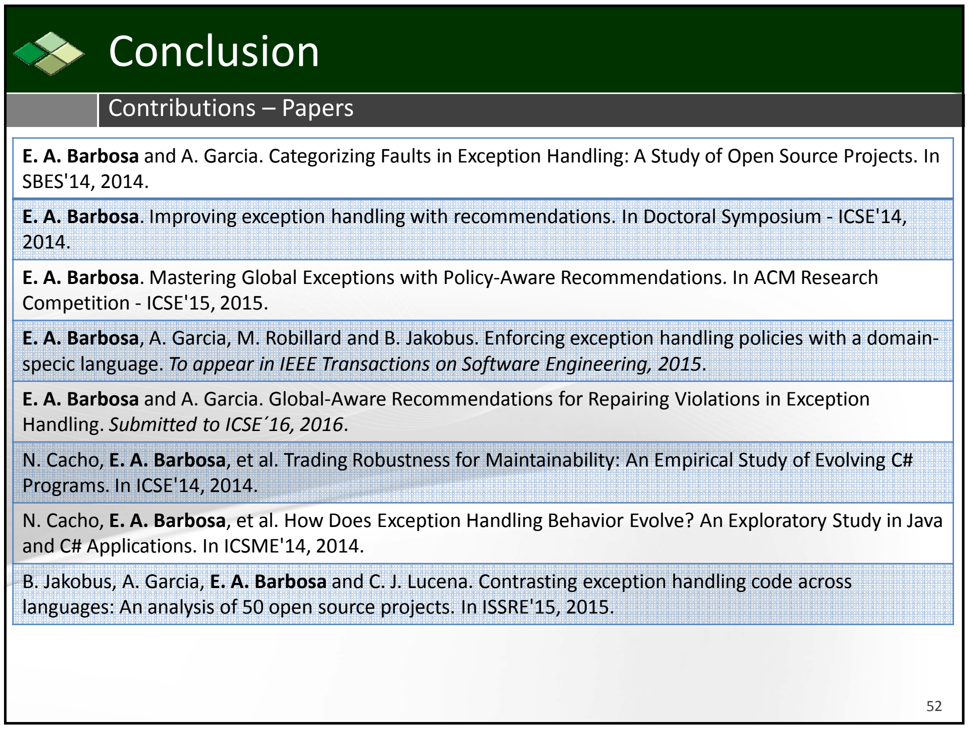


Conclusion

Contributions

- Recommender systems for software engineering:
  - “Speculative” analysis
  - Mitigating the “cold-start” problem
  - Experimental evaluation

November - 2015 51



Conclusion

Contributions – Papers

E. A. Barbosa and A. Garcia. Categorizing Faults in Exception Handling: A Study of Open Source Projects. In SBES'14, 2014.

E. A. Barbosa. Improving exception handling with recommendations. In Doctoral Symposium - ICSE'14, 2014.

E. A. Barbosa. Mastering Global Exceptions with Policy-Aware Recommendations. In ACM Research Competition - ICSE'15, 2015.

E. A. Barbosa, A. Garcia, M. Robillard and B. Jakobus. Enforcing exception handling policies with a domain-specic language. *To appear in IEEE Transactions on Software Engineering, 2015*.

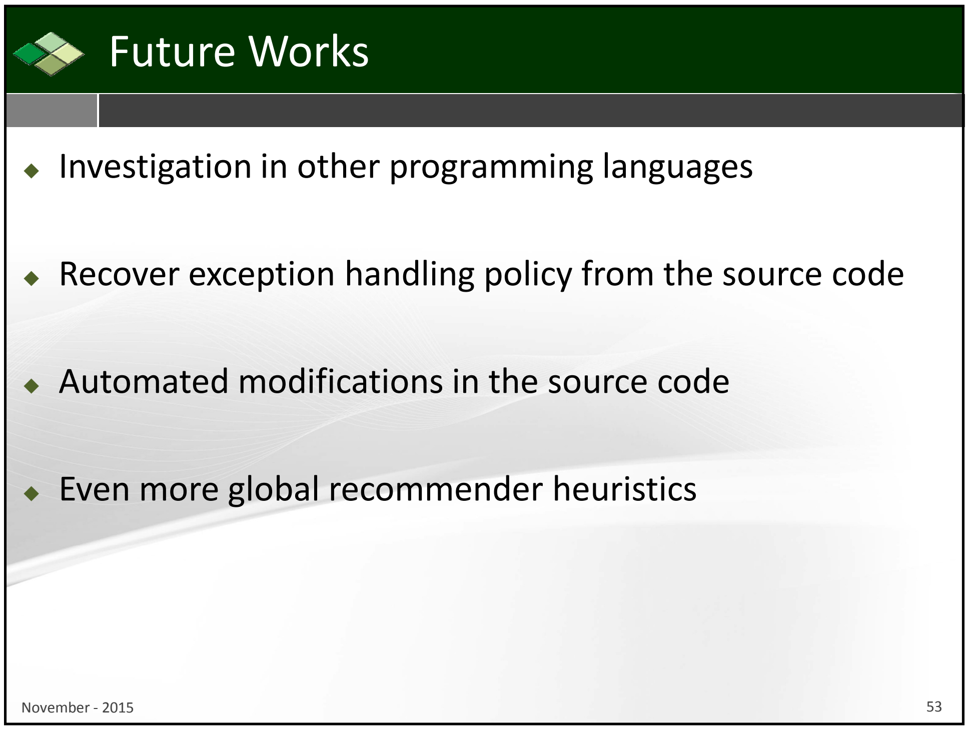
E. A. Barbosa and A. Garcia. Global-Aware Recommendations for Repairing Violations in Exception Handling. *Submitted to ICSE´16, 2016*.

N. Cacho, **E. A. Barbosa**, et al. Trading Robustness for Maintainability: An Empirical Study of Evolving C# Programs. In ICSE'14, 2014.

N. Cacho, **E. A. Barbosa**, et al. How Does Exception Handling Behavior Evolve? An Exploratory Study in Java and C# Applications. In ICSME'14, 2014.

B. Jakobus, A. Garcia, **E. A. Barbosa** and C. J. Lucena. Contrasting exception handling code across languages: An analysis of 50 open source projects. In ISSRE'15, 2015.

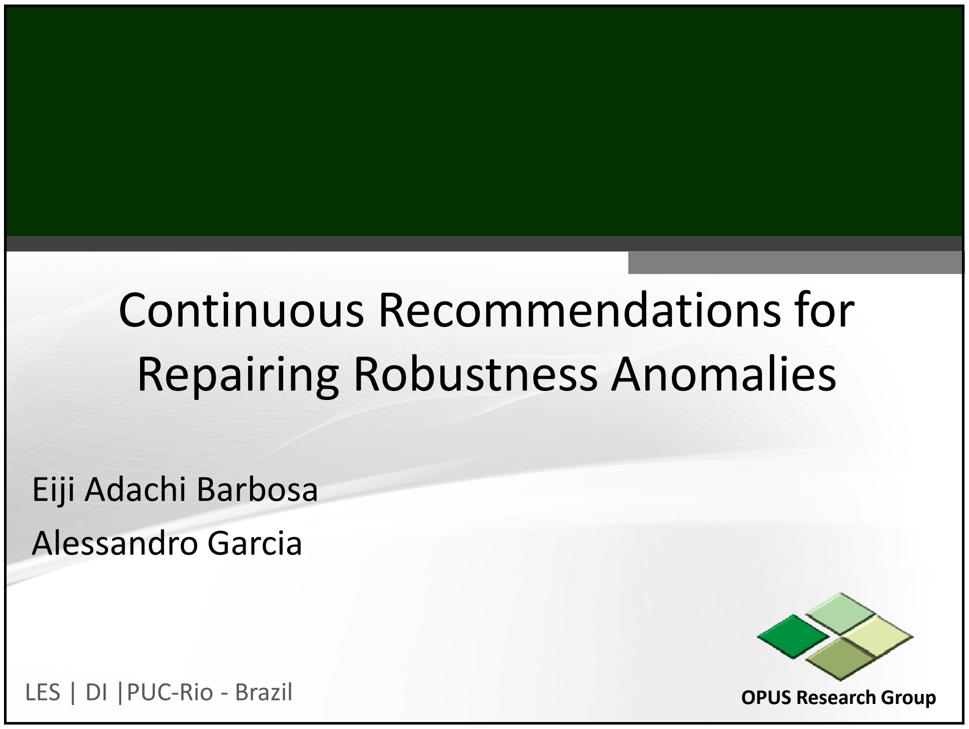
52



Future Works

- Investigation in other programming languages
- Recover exception handling policy from the source code
- Automated modifications in the source code
- Even more global recommender heuristics

November - 2015 53



Continuous Recommendations for Repairing Robustness Anomalies

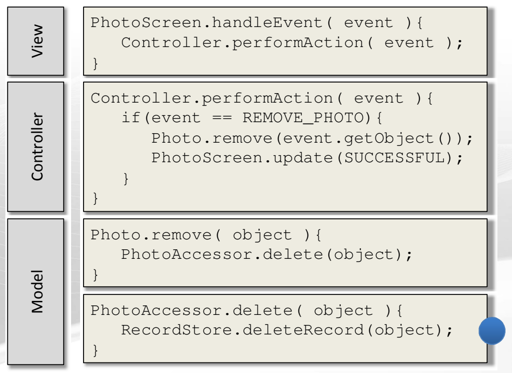
Eiji Adachi Barbosa
Alessandro Garcia

LES | DI |PUC-Rio - Brazil

OPUS Research Group

Lack of Explicit Exception Handling Policies

Problem Statement



```
PhotoScreen.handleEvent( event ){
   Controller.performAction( event );
}
```

```
Controller.performAction( event ){
   if(event == REMOVE_PHOTO){
      Photo.remove(event.getObject());
      PhotoScreen.update(SUCCESSFUL);
   }
}
```

```
Photo.remove( object ){
   PhotoAccessor.delete(object);
}
```

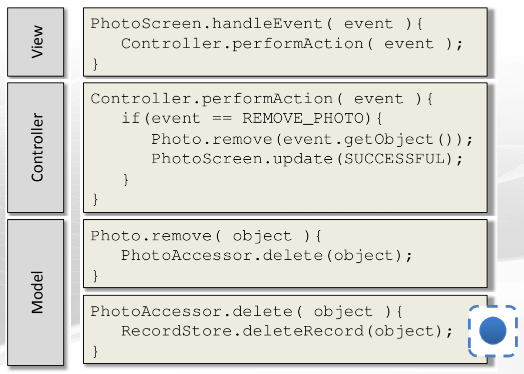
```
PhotoAccessor.delete( object ){
   RecordStore.deleteRecord(object);
}
```

- Leave exception unhandled
  - If exception occurs, program execution is terminated

November - 2015 55

Lack of Explicit Exception Handling Policies

Problem Statement



```
PhotoScreen.handleEvent( event ){
   Controller.performAction( event );
}
```

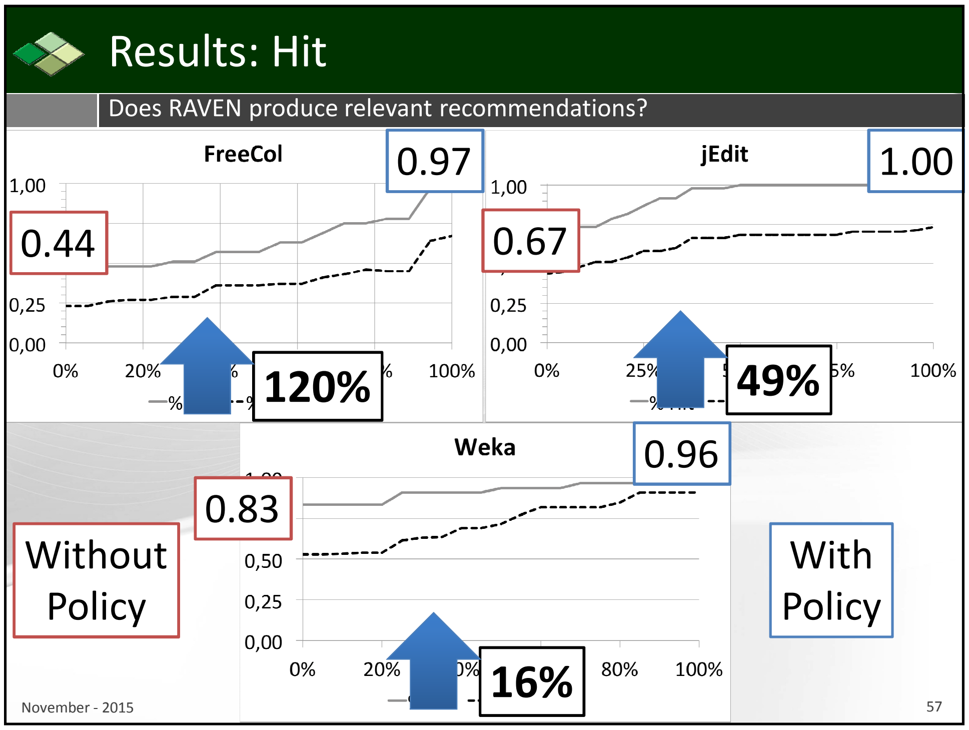
```
Controller.performAction( event ){
   if(event == REMOVE_PHOTO){
      Photo.remove(event.getObject());
      PhotoScreen.update(SUCCESSFUL);
   }
}
```

```
Photo.remove( object ){
   PhotoAccessor.delete(object);
}
```

```
PhotoAccessor.delete( object ){
   RecordStore.deleteRecord(object);
}
```

- Handle exception in the place where it occurs

November - 2015 56

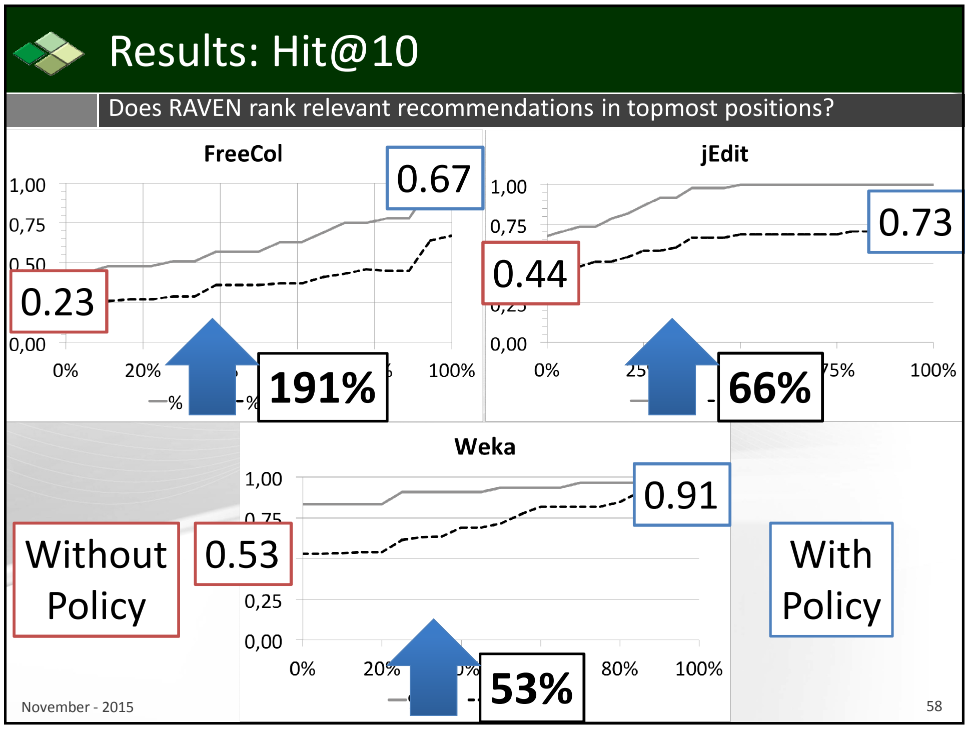


Results: Hit

Does RAVEN produce relevant recommendations?

November - 2015

57



Results: Hit@10

Does RAVEN rank relevant recommendations in topmost positions?

November - 2015

58