

Aspect Mining & Refactoring in Embedded Systems

Tom Tourwé

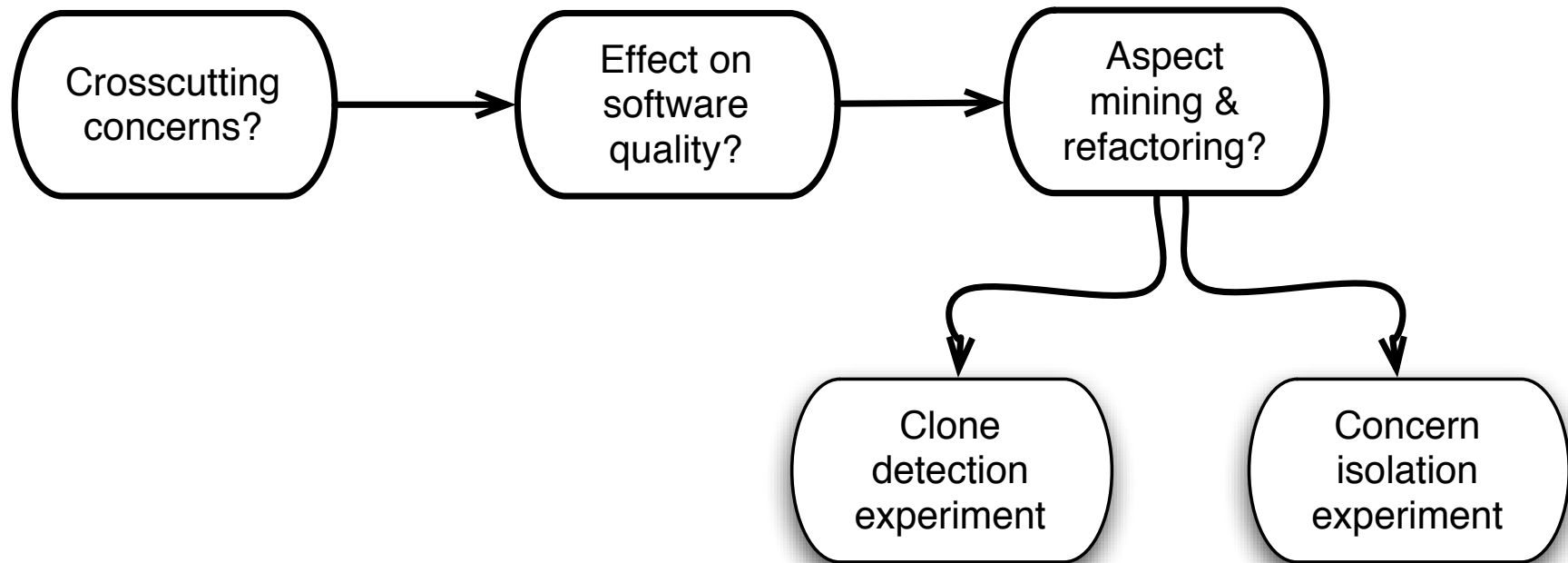
Interactive Software Development & Renovation

Software Engineering Cluster

C.W.I.

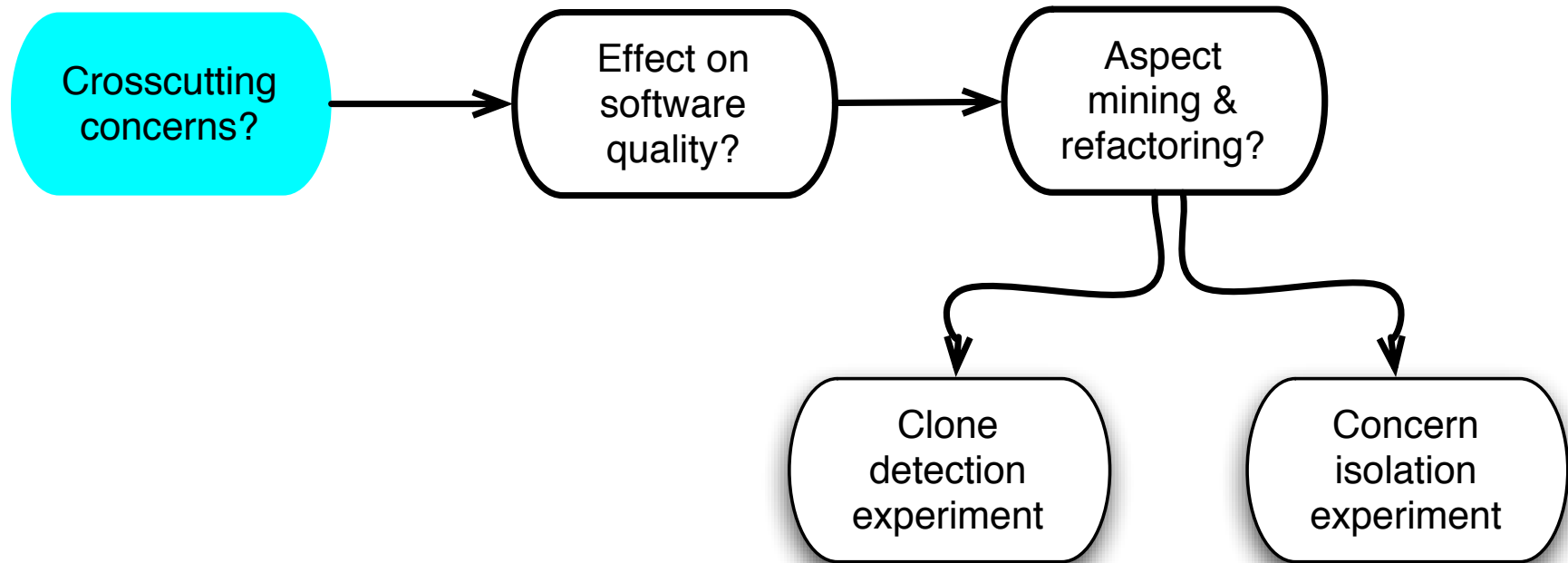
Overview

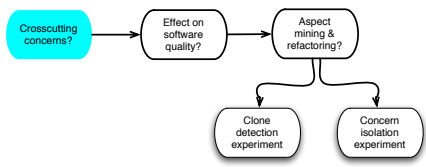
Improving software quality by improving the handling of “crosscutting concerns”



Overview

Improving software quality by improving the handling of “crosscutting concerns”

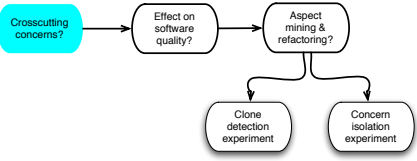




Separation of concerns

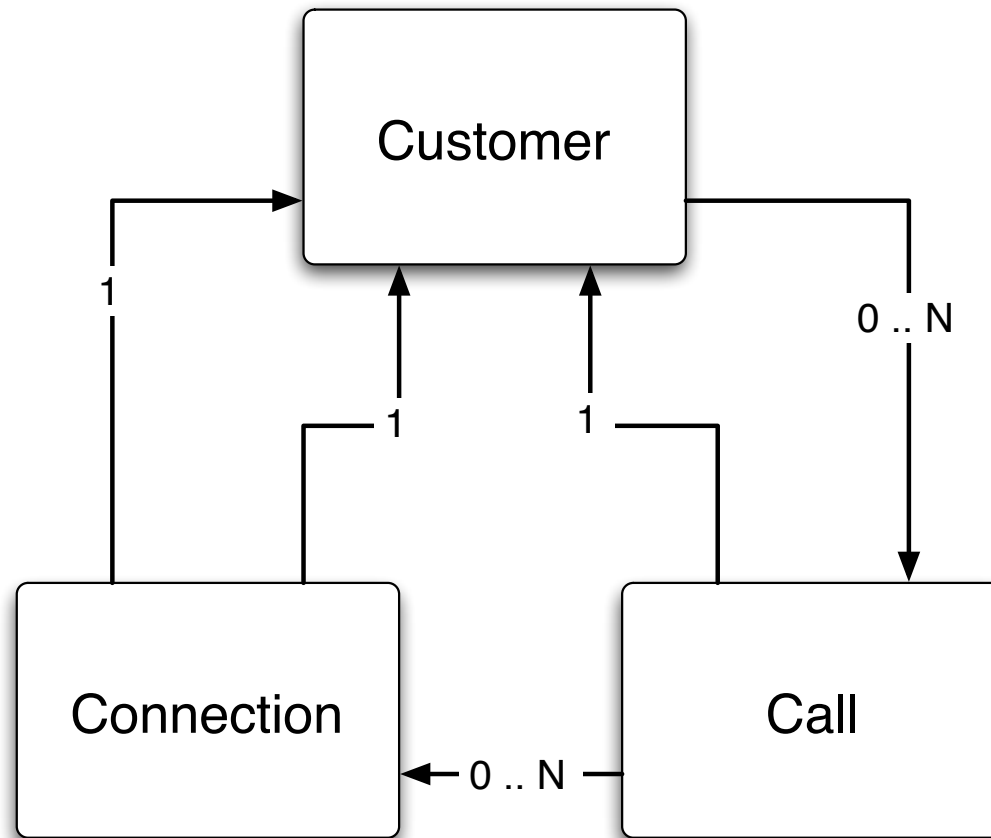
“Any given problem involves **different kinds of concerns**, which should be **identified** and **separated** to cope with **complexity** and to achieve the required engineering **quality factors**, such as adaptability, maintainability, extendibility and reusability”

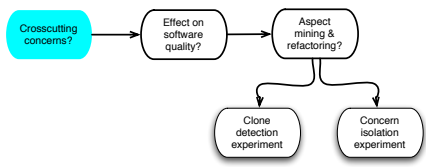




Example

“Core” concerns in a simple Telecom application

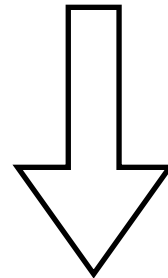




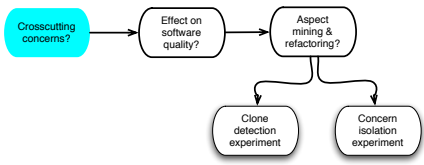
What are crosscutting concerns?

But

- only one structure is implemented
- structure for one concern is not necessarily suited for other concerns

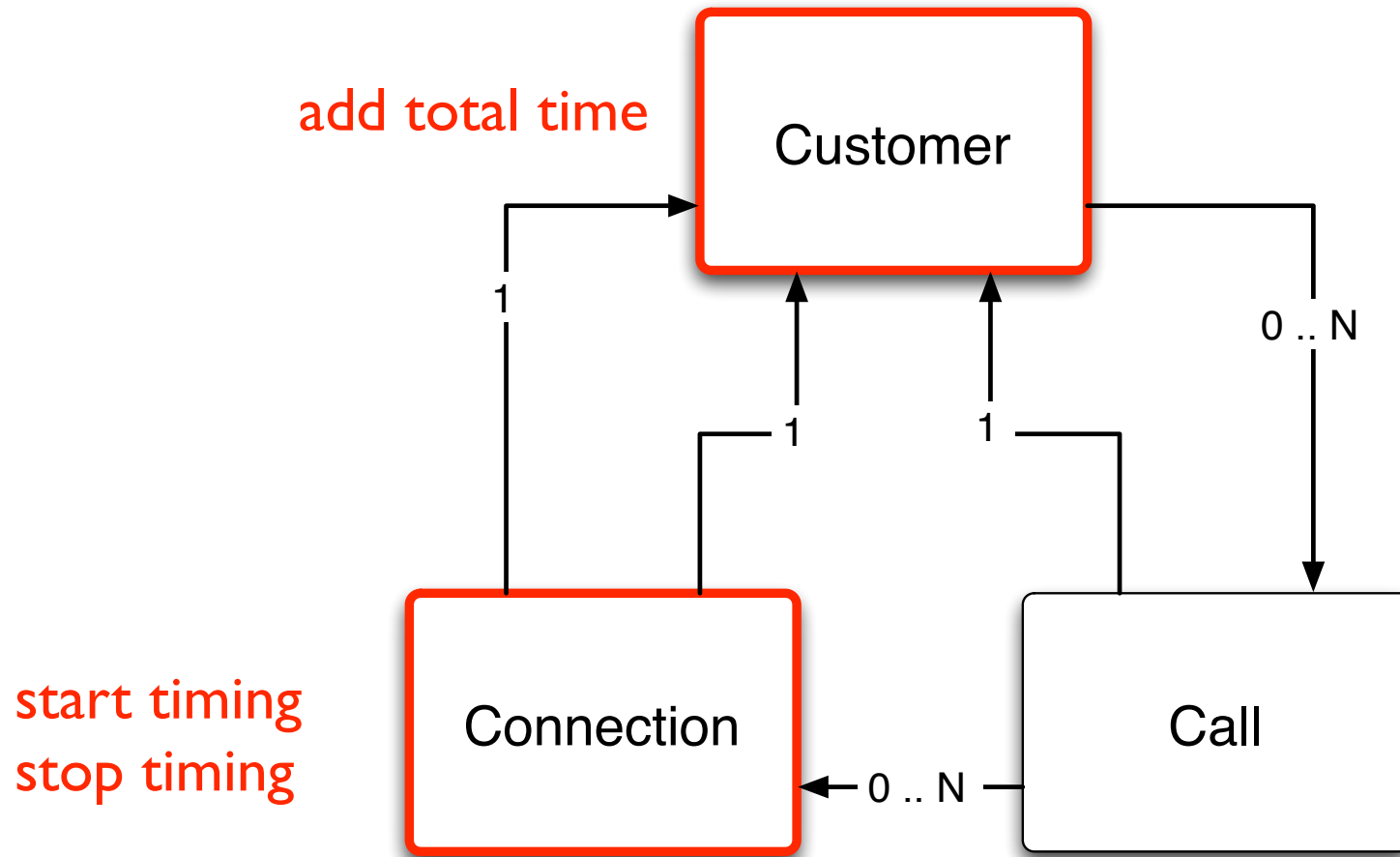


some concerns “crosscut” the chosen structure

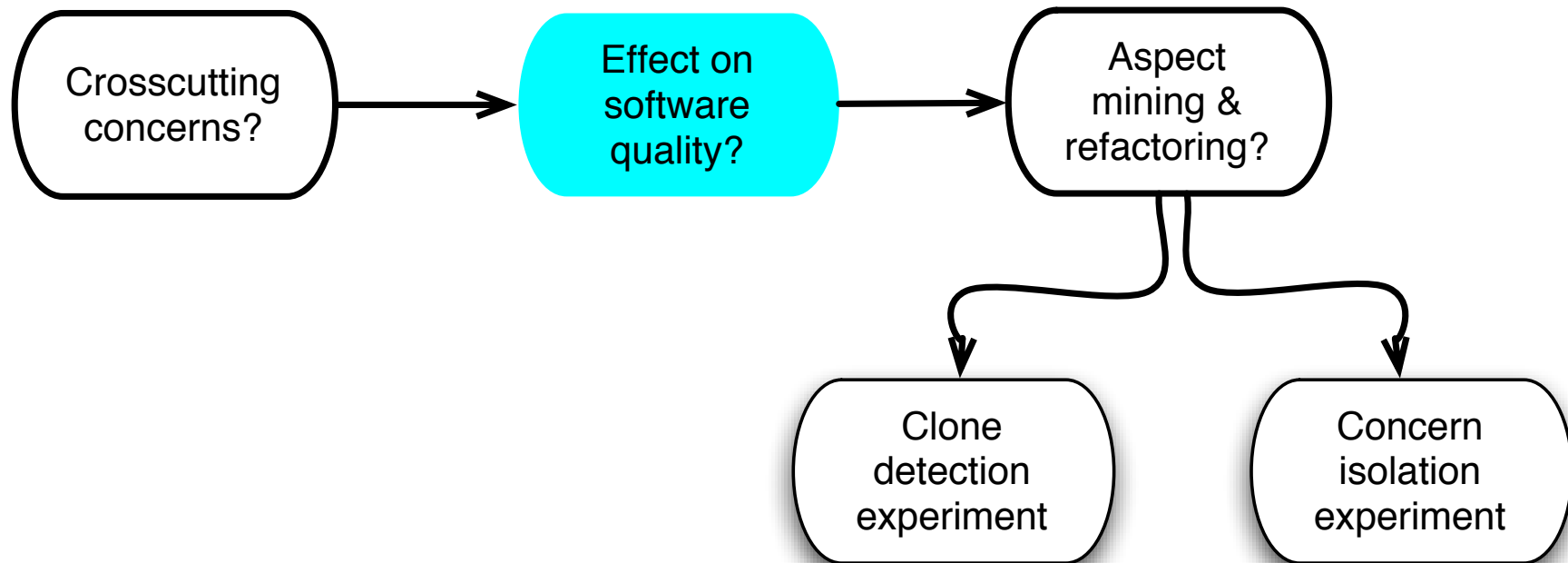


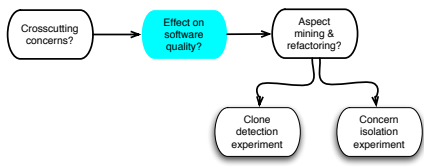
Example

“Crosscutting” concern in a simple Telecom application: **Timing**



Overview

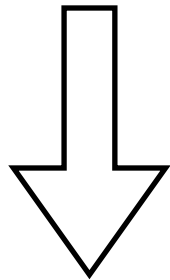




Effect of crosscutting concerns

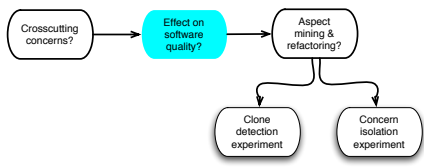
“Crosscutting concerns lead to scattering and tangling”

code size: 16.406 LoC
 concern size: 4182 LoC



25.5% of code is
 crosscutting concern code





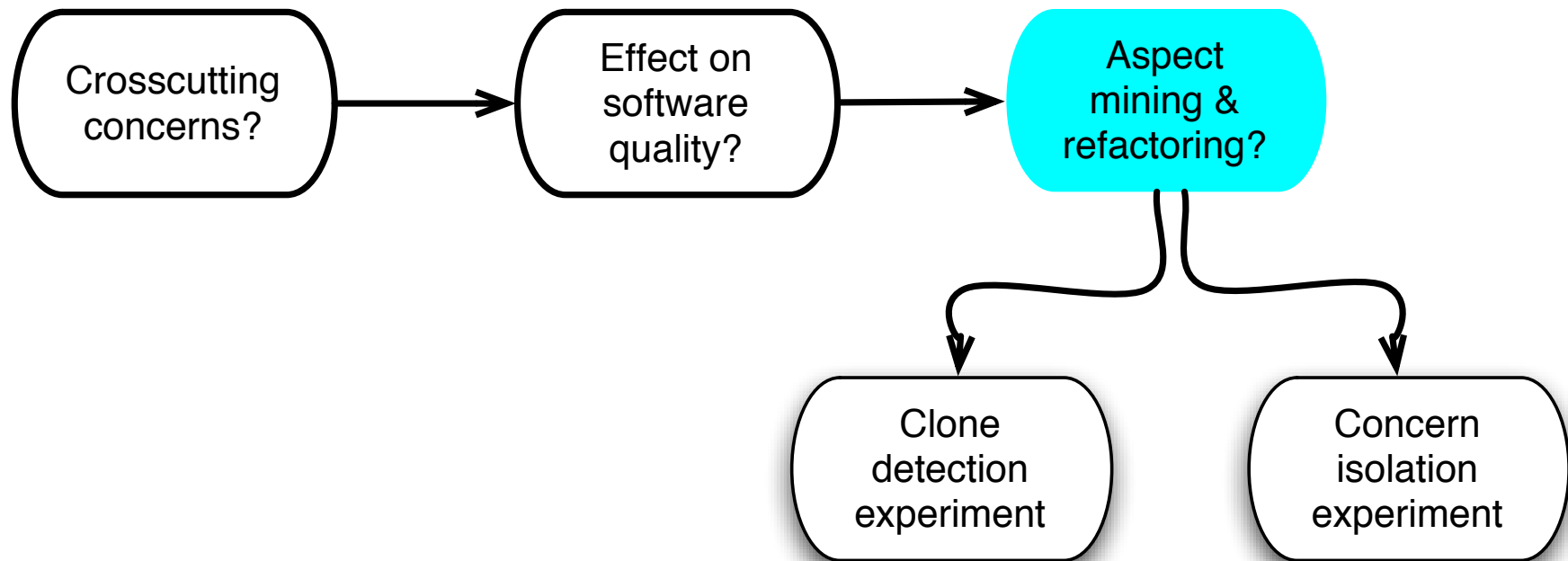
Effect of crosscutting concerns

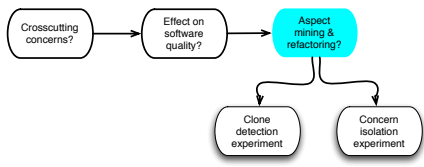
“Crosscutting concerns lead to errors and inconsistencies”

experimental observations for
“null-checking” concern

	deviation s	unintended	intended
CC1 (3kLoC)	8	0	8
CC2 (16kLoC)	65	58	7
CC3 (12kLoC)	23	16	7

Overview





Aspect mining & refactoring

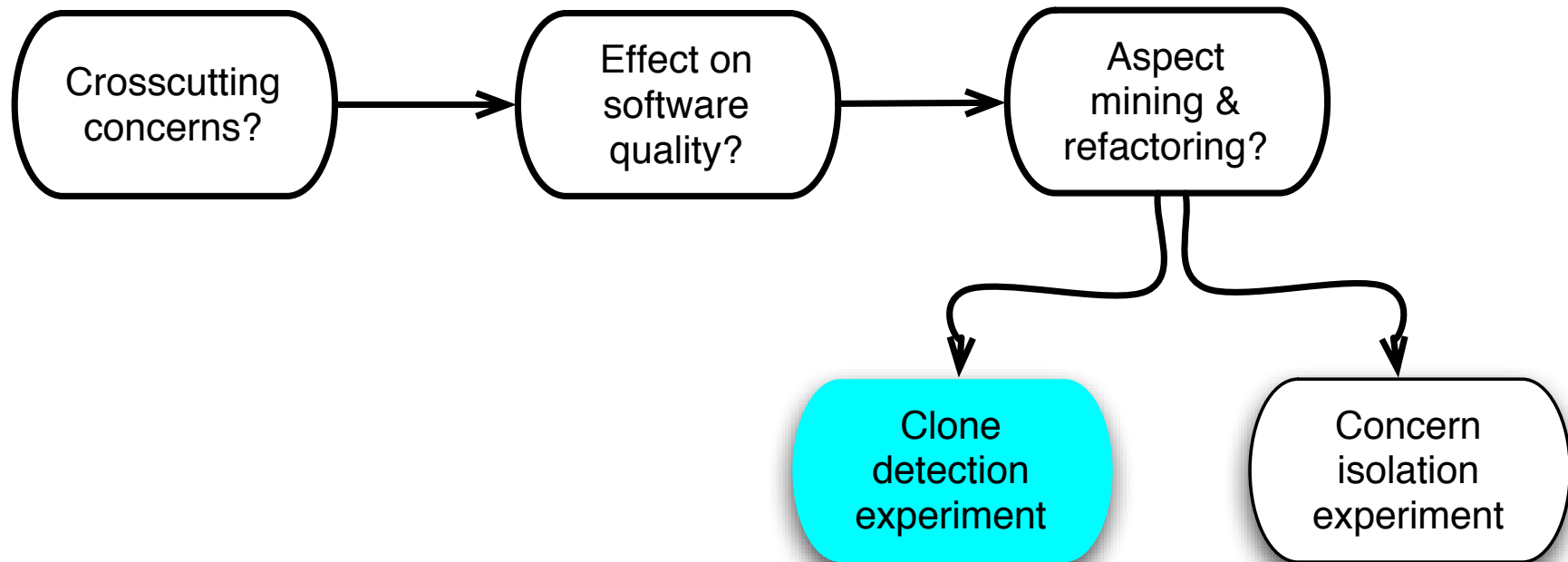
Aspect mining

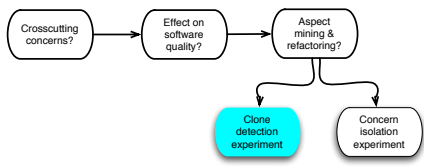
the process of (automatically) identifying crosscutting concerns in an existing software system's source code

Aspect refactoring

the process of (automatically) transforming crosscutting concerns to *aspects*

Overview





Clone detection experiment

Motivation

Assess suitability of clone detection for identifying crosscutting concerns

Experiment

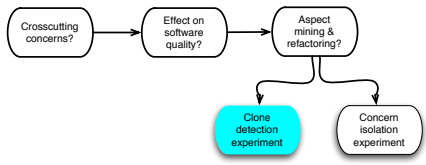
5 crosscutting concerns

3 clone detectors

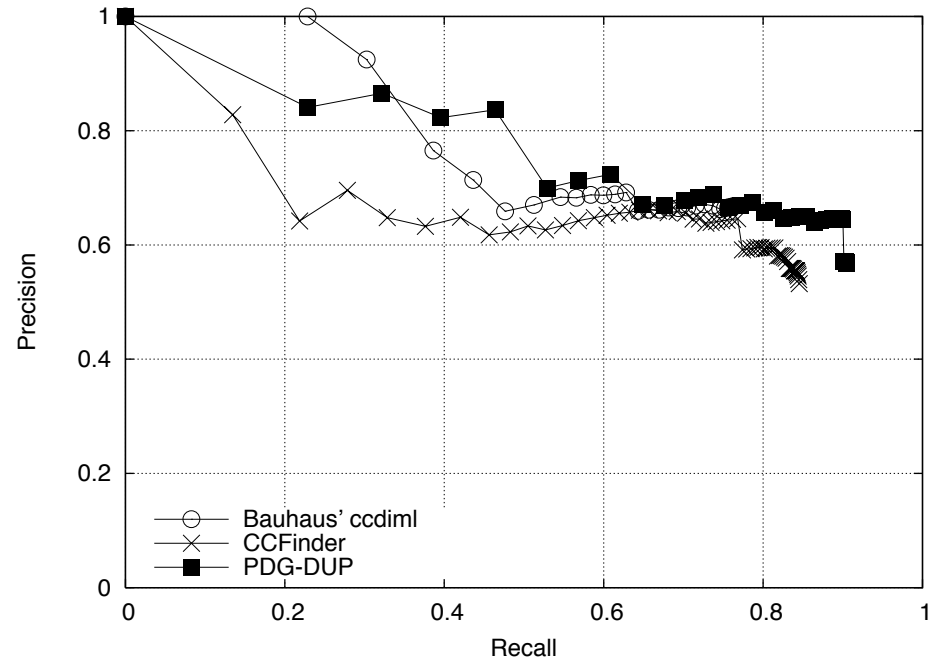
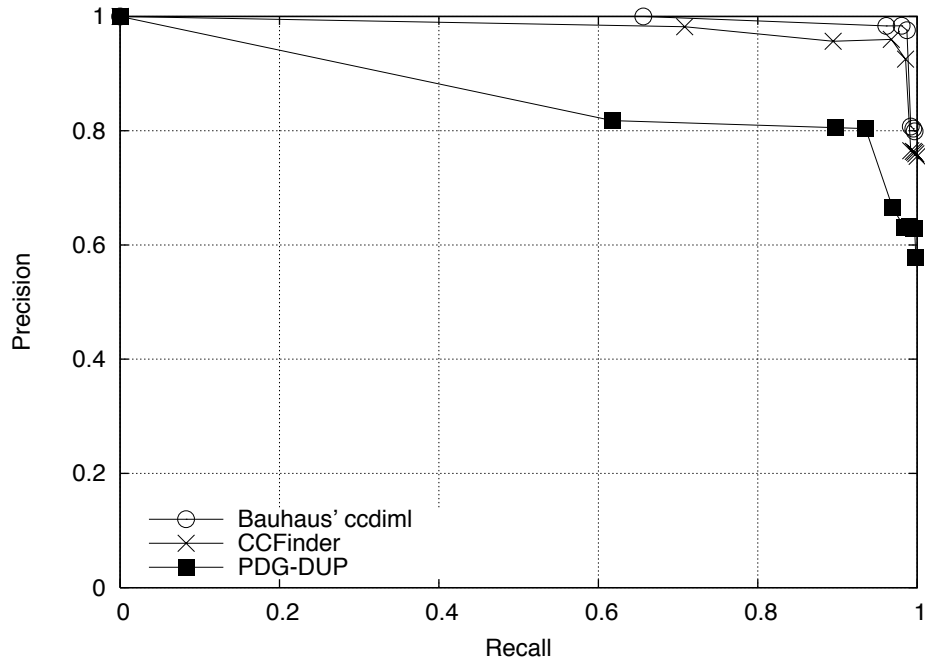
measure precision and recall



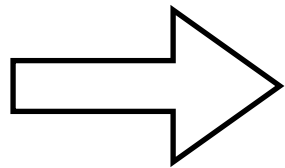
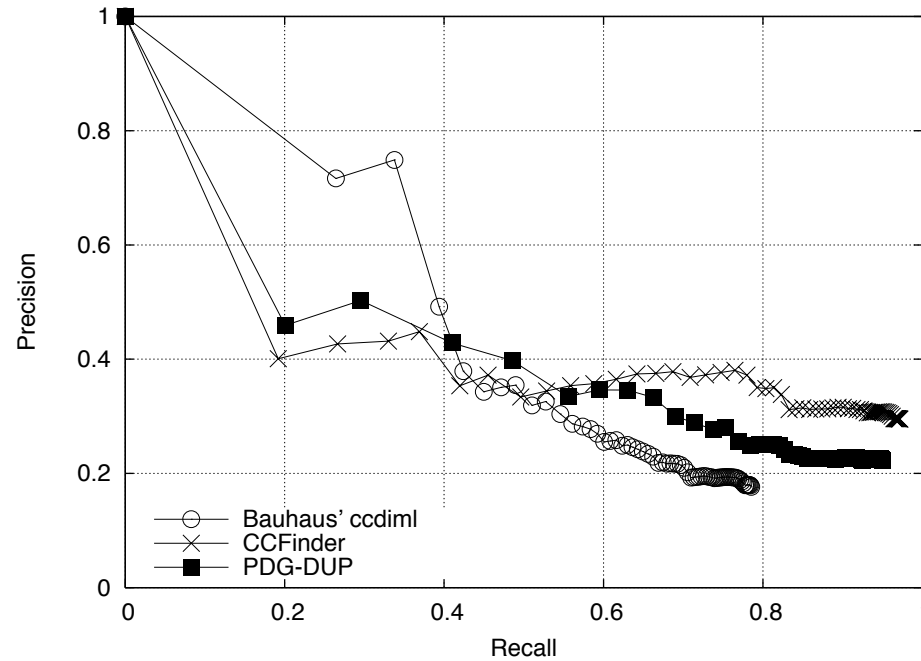
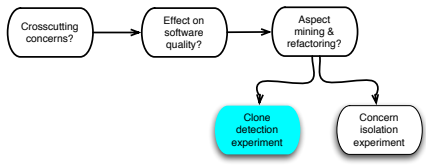
M.Bruntink, A. van Deursen, R. van Engelen and T. Tourwé, “An evaluation of clone detection techniques for identifying crosscutting concerns”, 2004.



Some concerns covered very well

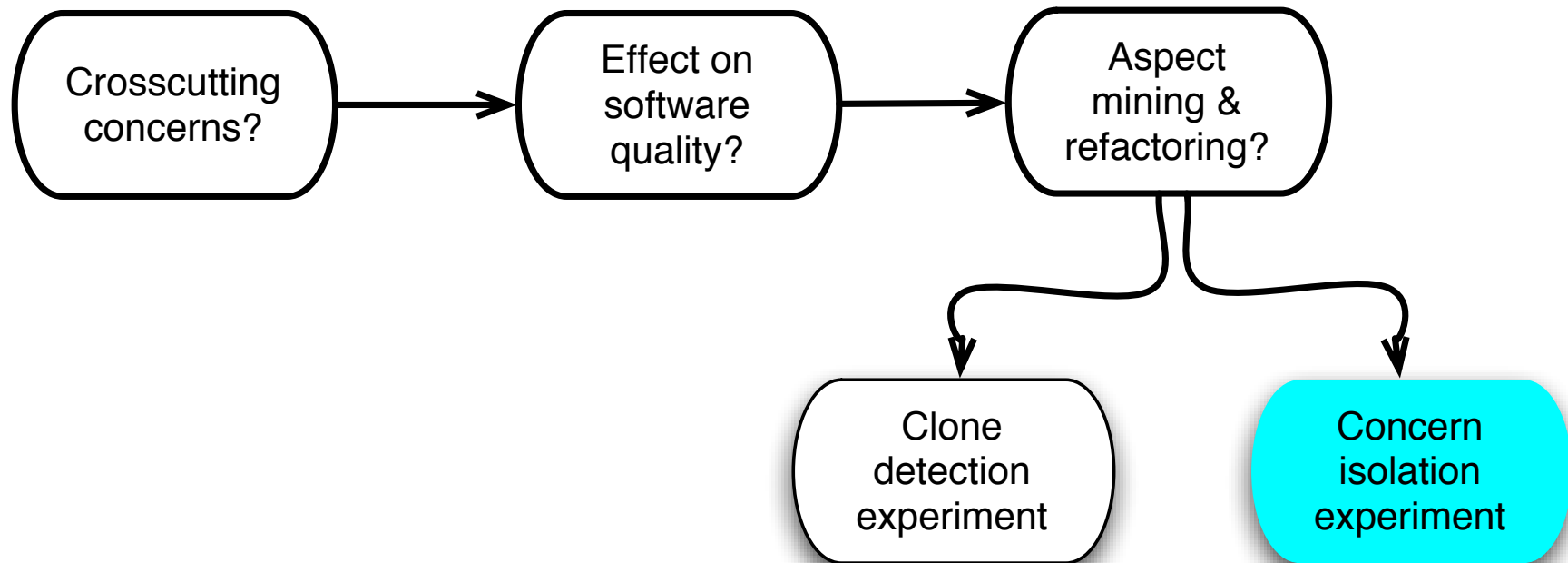


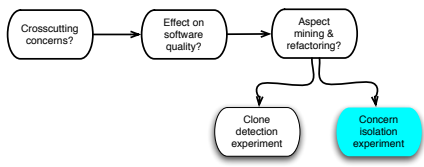
Other concerns covered badly



primarily due to high *tangling*

Overview





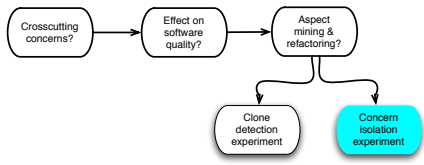
Concern Isolation Experiment

Motivation

Reduce code duplication and code size, and improve consistency and correctness by using aspects

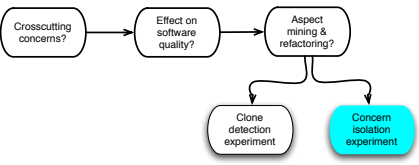
Experiment

Refactor I concern and qualitatively compare current approach to new approach



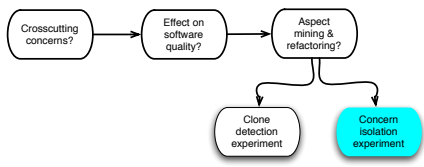
Effect on code size

	CCI	CC2	CC3
concern code	56	961	456
aspect code	46	132	787



Aspects reduce scattering & tangling

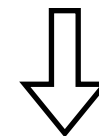




Aspects promote uniformity

strings used when logging an error

	Without aspects	With aspects
CC1	2	1
CC2	30	3
CC3	15	1



even these strings are uniform!

Conclusion

- Crosscutting concerns have a negative impact on software quality ...
- ... which can be reduced by using aspects ...
- ... which requires aspect mining & refactoring techniques ...
- ... which is what we are investigating.