Traceability Reuse for Change Impact Analysis in a Safety-critical System

MARKUS BORG, LUND UNIVERSITY

Markus Borg – Safety and Traceability Highlights

MSc Eng. Computer Science & Eng. 2002-2007

Development engineer, ABB 2007-2010

- Process automation (IEC 61511)
- SIL 2 (IEC 61508)
- Editor and compiler development

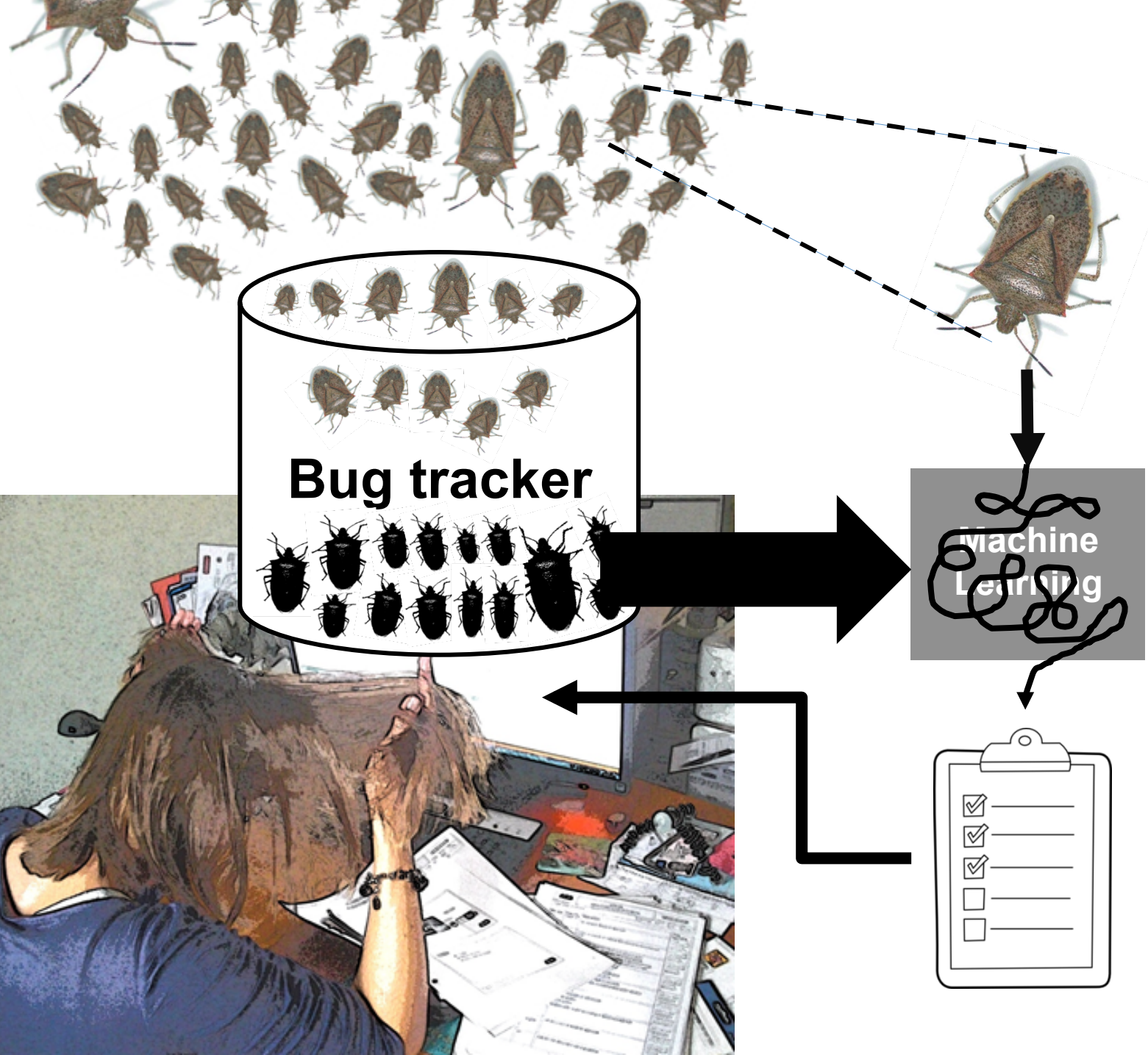
PhD, Lund University 2010-2015

- Empirical software engineering

Research interests

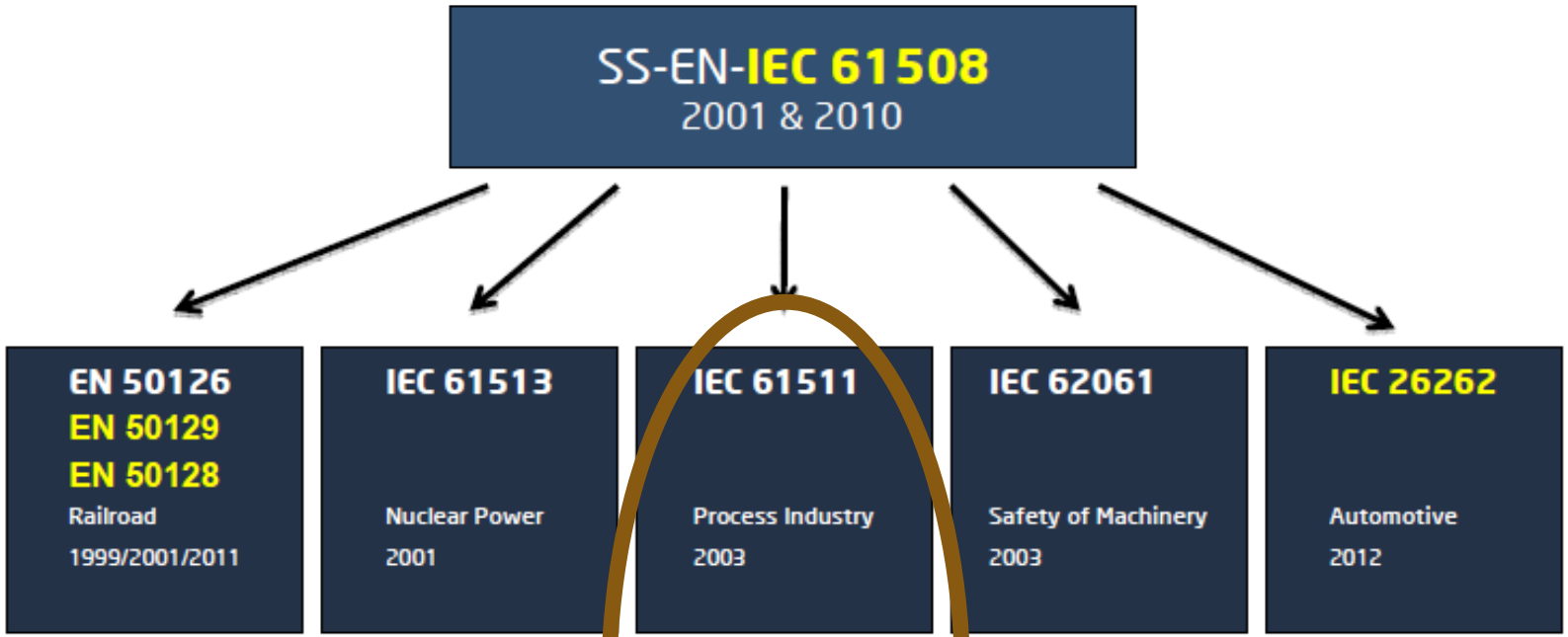
- Issue tracking and management
- Traceability and change impact analysis
- Certification of safety-critical software

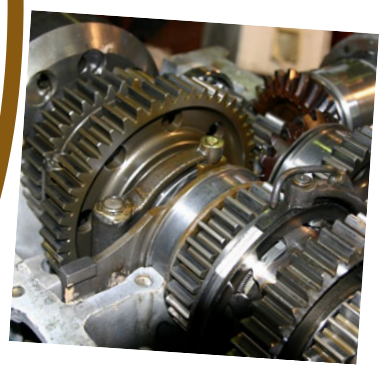
ERICSSON ABB SONY Qlik AXIS COMMUNICATIONS EASE


Bug tracker
Machine Learning


The Challenge
The Solution
The Evaluation

Background and Case Description





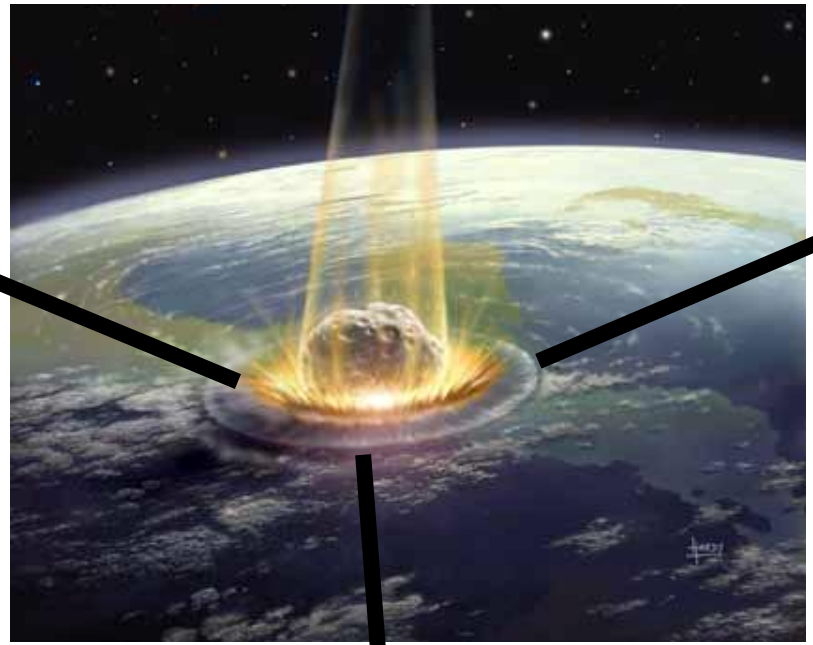
System under Study

- Evolution of large automation system (IEC 61511)
- Change impact analysis mandated by IEC 61508
- Developers put much effort into tracing impact
  - Mainly to comply with standards
- Could the developers benefit more from the tracing effort?

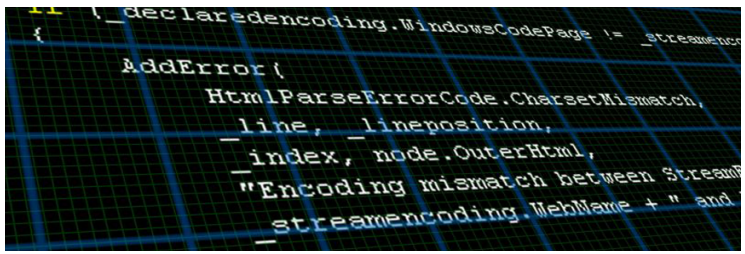
LUND UNIVERSITY

Change Impact Analysis

Reqs.



Tests



LUND UNIVERSITY

Change Impact Analysis Questions	
1	Is the reported problem safety critical?
2	In which versions/revisions does this problem exist?
3	How are general system functions and properties affected by the change?
4	**List modified code files/modules and their SIL classifications, and/or affected safety safety related hardware modules.**
5	**Which library items are affected by the change?** (e.g., lib , firmware functions, HW types, HW libraries)
6	**Which documents need to be modified?** (e. ., product requirements specifications, architecture, functional requ pecifications, design descriptions, schematics, functional test escriptions, design test descriptions)
7	**Which test cases need to be executed?** gn tests, functional tests, sequence tests, environmental/EMC tes s, FPGA simulations)
8	**Which user documents, including online help, n nodified?**
9	How long will it take to correct the problem, and ver y the correction?
10	What is the root cause of this problem?
11	How could this problem been avoided?
12	**Which requirements and functions need to be retested b l- uct test/system test organization?**

Challenges in IA: Findings from a Case Study

Interviews with 14 engineers in Sweden and India

- Intermediate results, not yet published

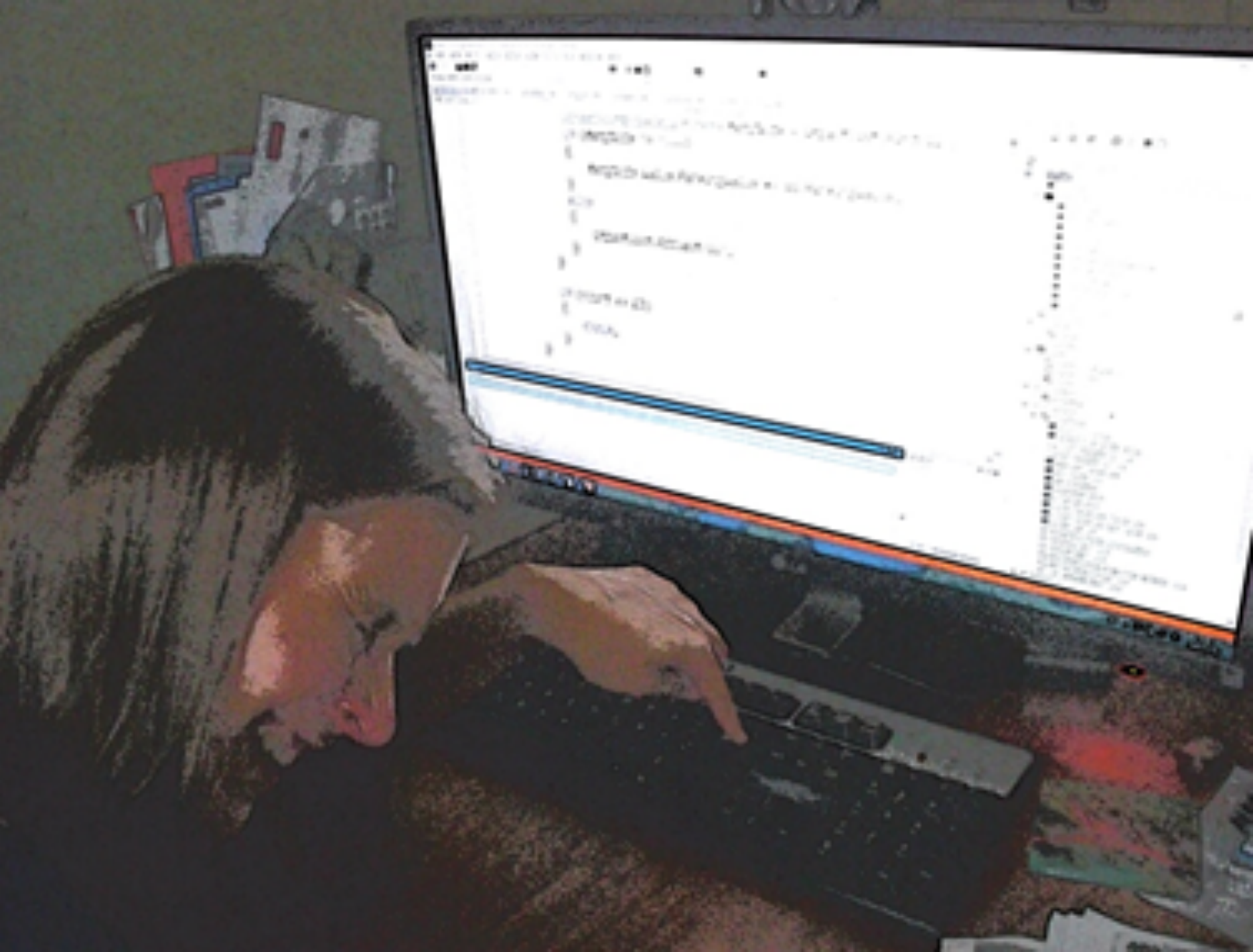
- Reporting how requirements are impacted
  - Developers much better at finding code impact
- Several developers question the value of rigorous IA
  - Spending hours on tracing just to comply with external assessment?
- Finding a balance in how much impact to report
  - Developers often either miss side-effects or report far too many

Challenges in IA: Findings from a Survey

- Survey with 97 respondents working with safety development
  - Technical report available (link)
  - Aerospace (36 %), Automotive (13 %), Railway (11 %)
  - Europe (56 %) US (28 %), Asia (8 %)
- Respondents assessed 13 challenges from the literature

1. Insufficient tool support
2. Estimating effort required to make a change
3. Vast number of artifacts to trace

LUND UNIVERSITY



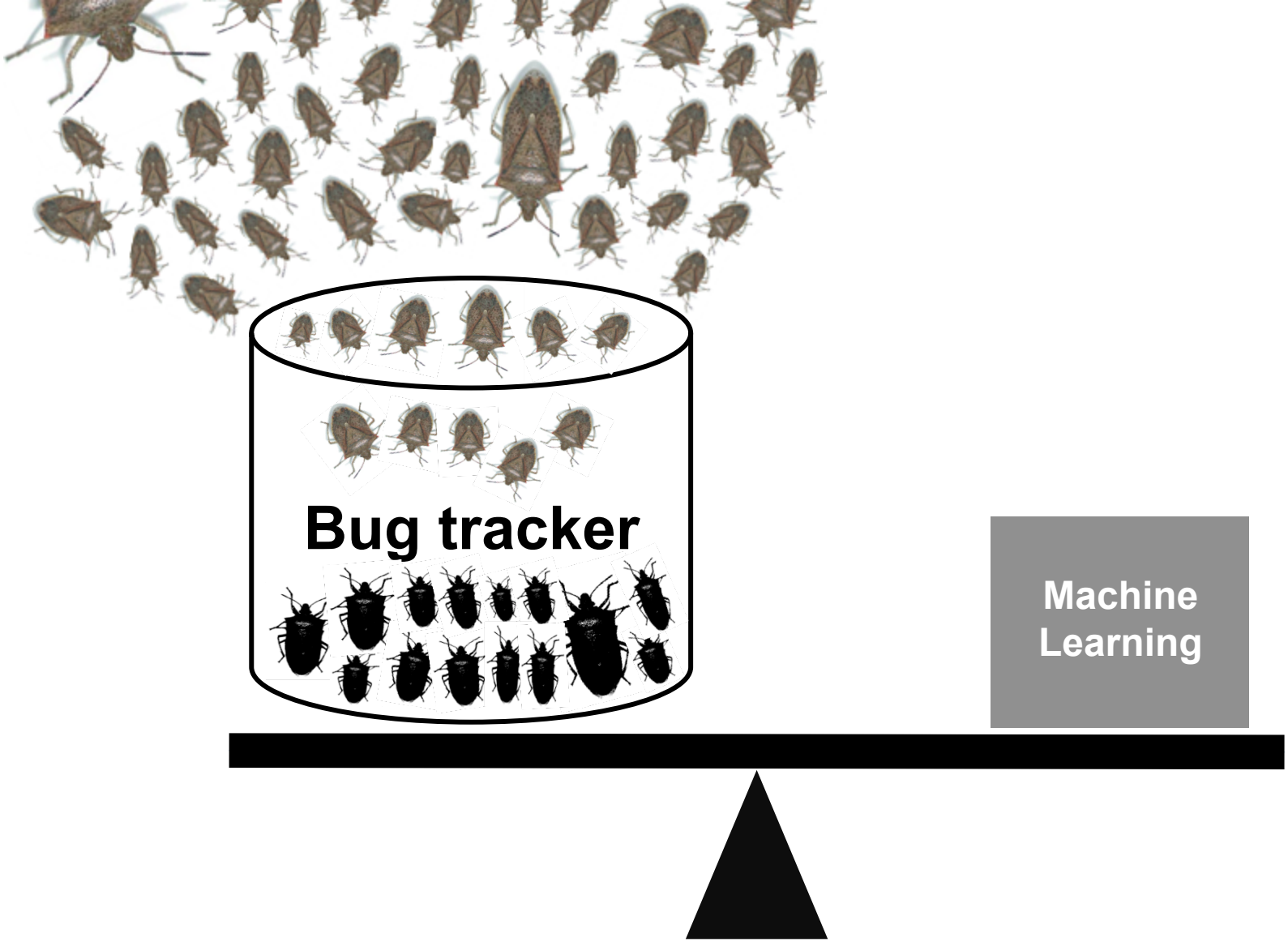
The Challenge

The Solution

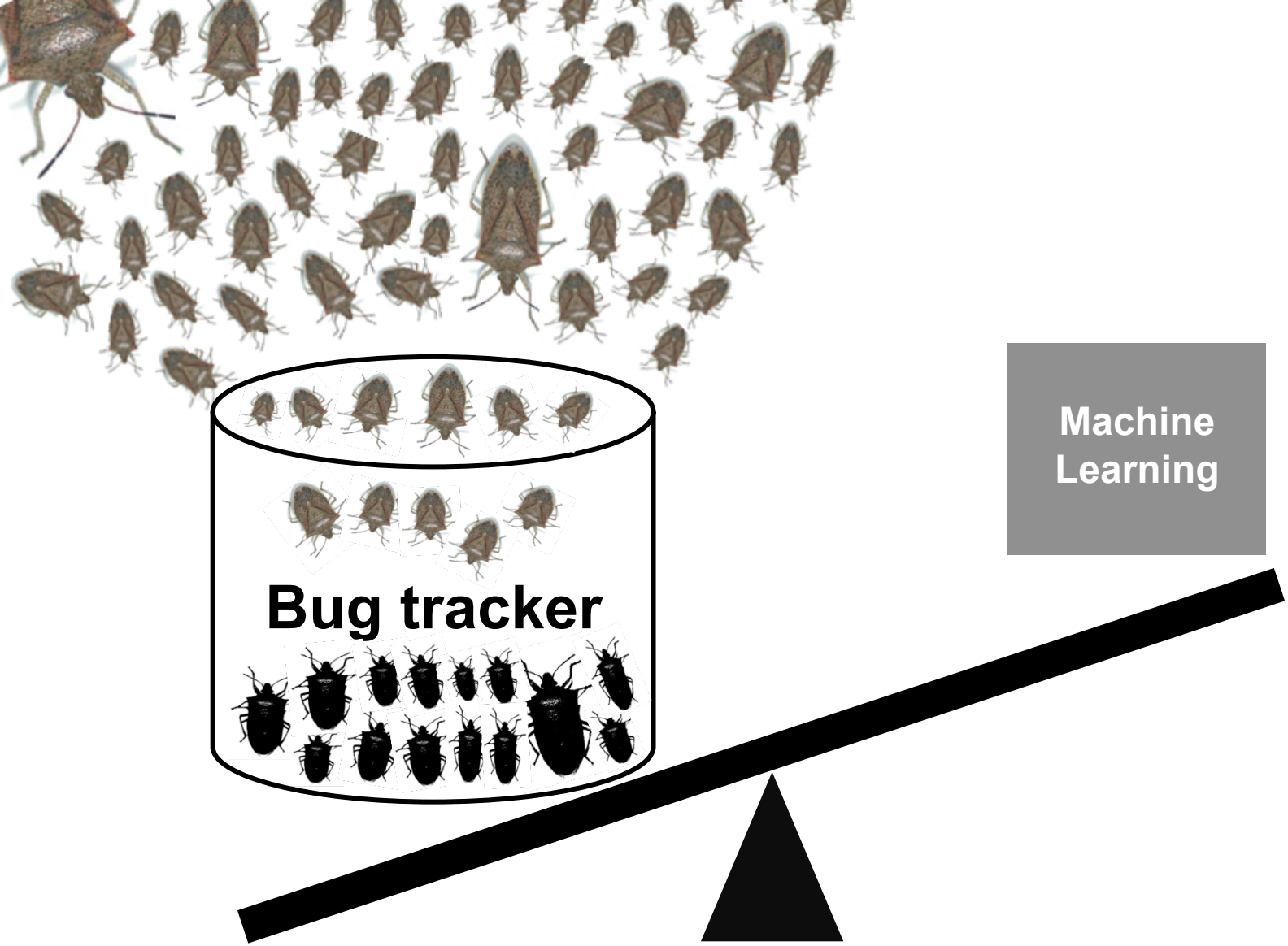
The Evaluation

LUND UNIVERSITY

LUND UNIVERSITY



LUND UNIVERSITY



Machine Learning

LUND UNIVERSITY

Decision Support for Impact Analysis

- Goal:

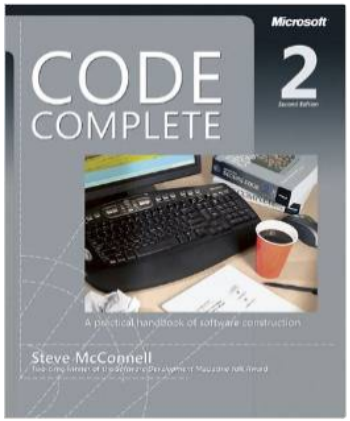
Intuitive tool to jump start analyses based on historical data

Faster + more accurate analyses compared to fully manual work

- Approach

Step 1: Mine the project history

Step 2: Recommend impact for new bug fix



Flip to back

See all 2 images

Code Complete: A Practical Handbook of Software Construction, Second Edition

by Steve McConnell (Author) Paperback – June 19, 2004

★★★★½ 219 customer reviews

ISBN-13: 079-0145196705 | ISBN-10: 0735619670 | Edition: 2nd

Buy New
Price: $31.72

Rent
Price: $22.00

61 New from $28.00 | 56 Used from $18.95

	Amazon Price	New from	Used from
Kindle	$24.79	—	—
Paperback	$31.72	$28.00	$18.95
Unknown Binding	—	$53.53	$70.87

Widely considered one of the best practical guides to programming, Steve McConnell's original CODE COMPLETE has been helping developers write better software for more than a decade. Now this classic book has been fully updated and revised with leading-edge practices—and hundreds of new code samples—illustrating the art and science of software construction. Capturing the body of knowledge available from research, academia, and everyday commercial practice, McConnell

Read more

Share

Rent $22.00

Buy New $31.72

Qty: 1 List Price: $49.99
Save: $18.27 (37%)

FREE Shipping on orders over $35.

In Stock.
Ships from and sold by Amazon.com.
Gift-wrap available.

Yes, I want FREE Two-Day Shipping with Amazon Prime

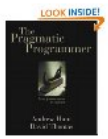
Add to Cart

Sign in to turn on 1-click ordering

Want it tomorrow, Sept. 24? Order within **9 hrs 43 mins** and choose **One-Day Shipping** at checkout. Details

Add to Wish List

Customers Who Bought This Item Also Bought



The Pragmatic Programmer: From ...
› Andrew Hunt
★★★★½ (255)
Paperback
$37.44 Prime

Clean Code: A Handbook of Agile Software ...
› Robert C. Martin
★★★★½ (194)
Paperback
$38.85 Prime

Design Patterns: Elements of Reusable ...
› Erich Gamma
★★★★½ (366)
Hardcover
$45.76 Prime

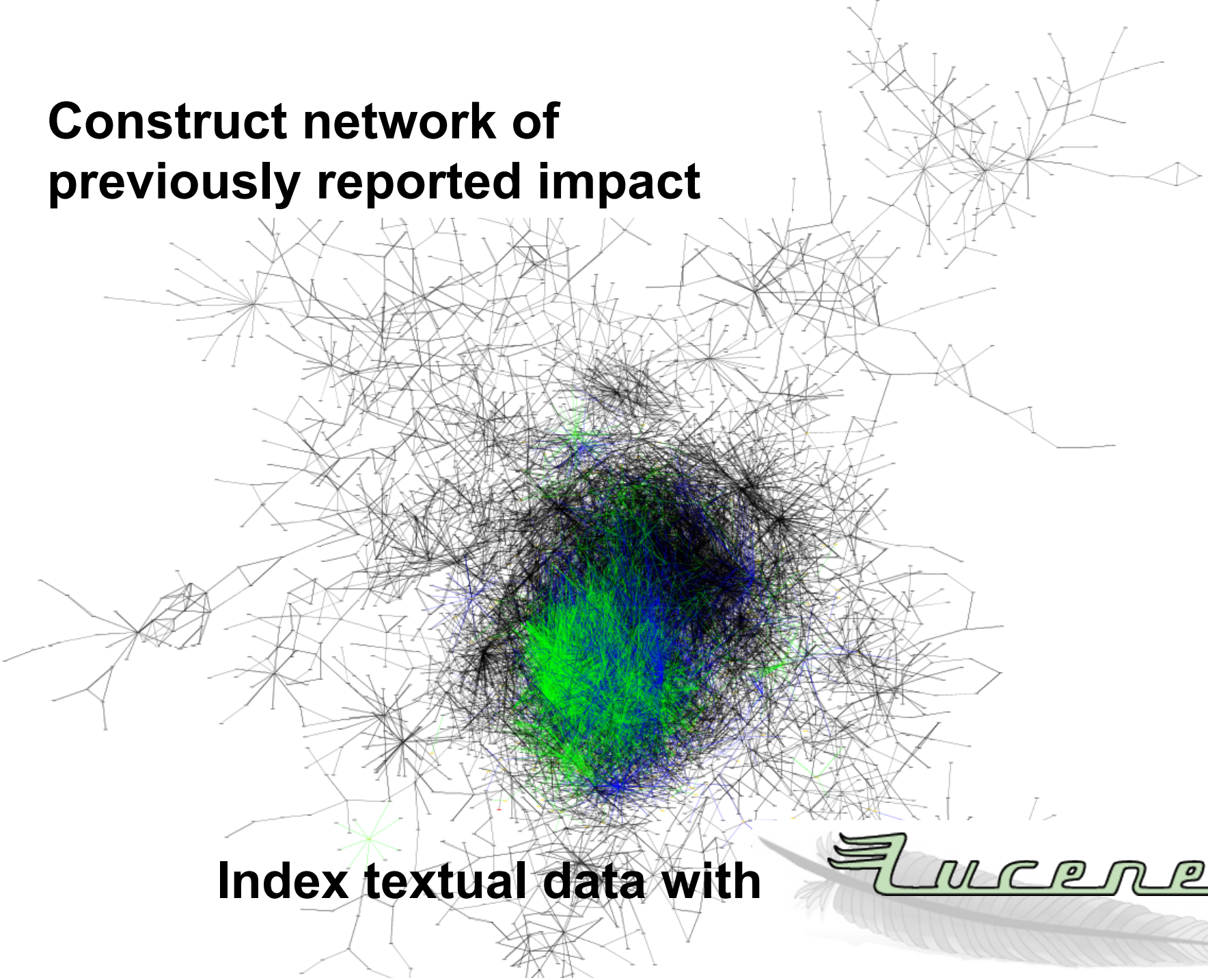
The Mythical Man-Month: Essays on Software ...
Frederick P. Brooks Jr.
★★★★½ (222)
Paperback
$30.26 Prime

Refactoring: Improving the Design of Existing ...
› Martin Fowler
★★★★½ (180)
Hardcover
$48.57 Prime

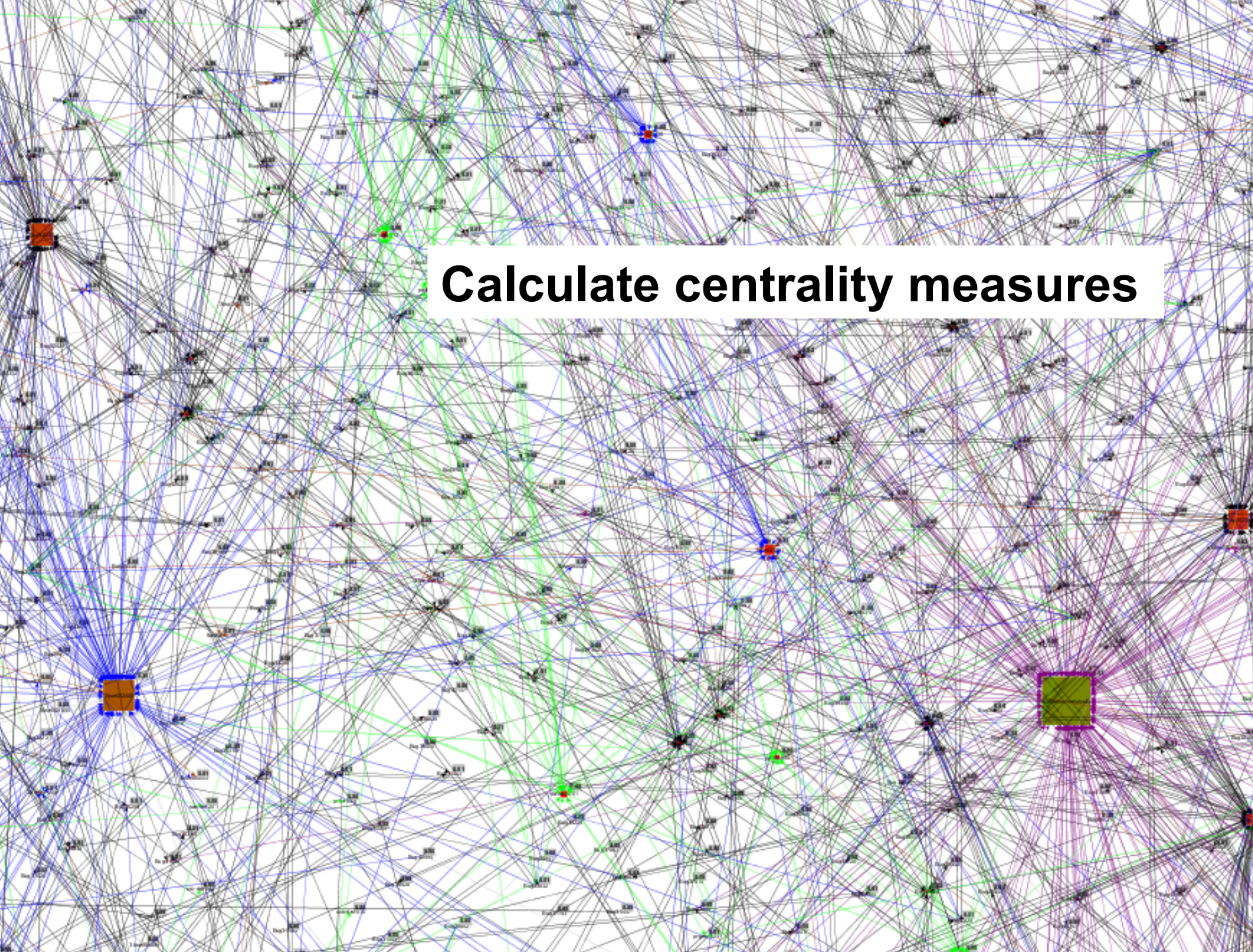
Head First Design Patterns
› Eric Freeman
★★★★½ (431)
Paperback
$37.99 Prime

Programming Pearls (2nd Edition)
Jon Bentley
★★★★½ (63)
Paperback
$31.75 Prime

Construct network of previously reported impact

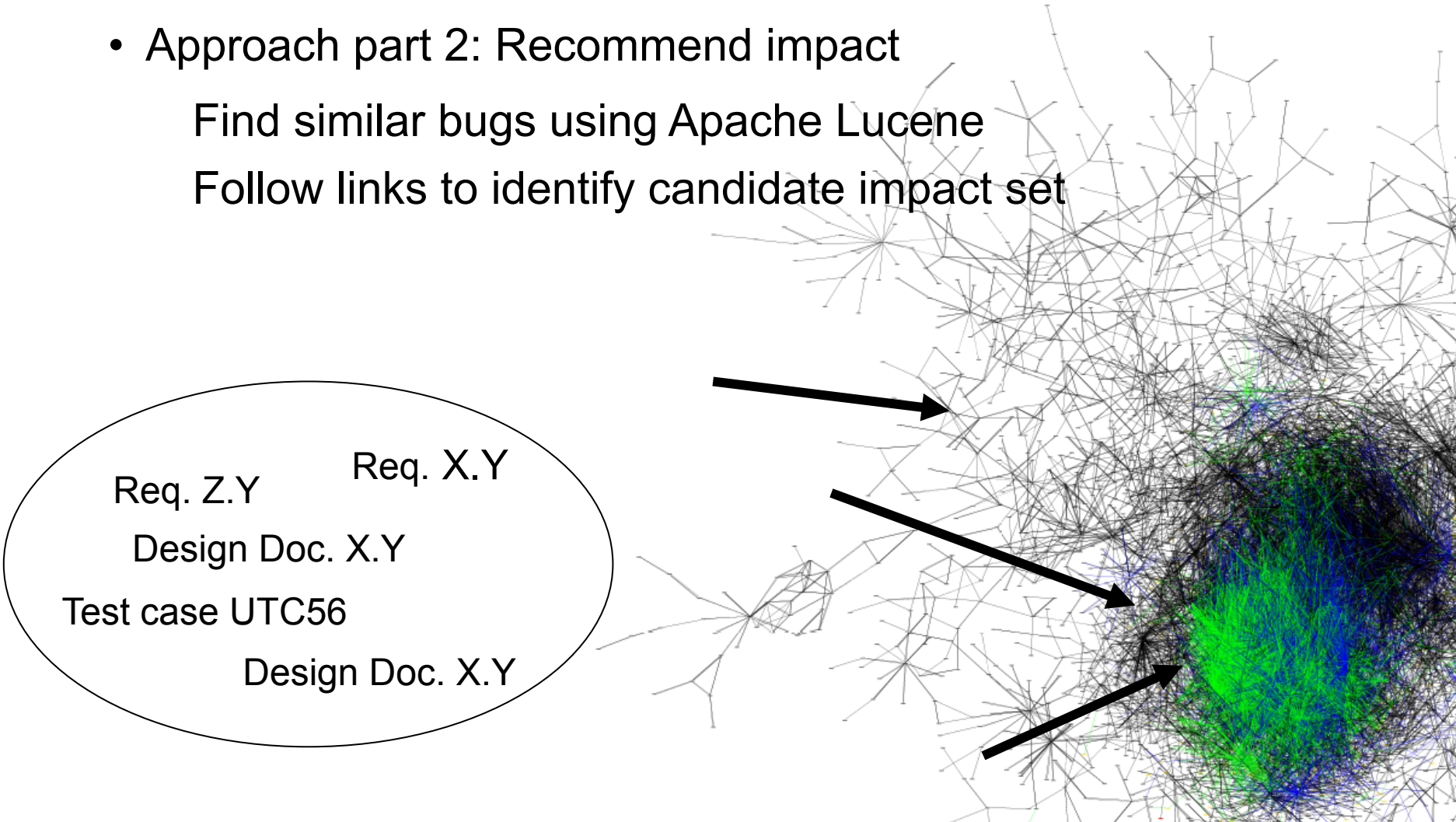


Index textual data with


Calculate centrality measures

Decision Support for Impact Analysis

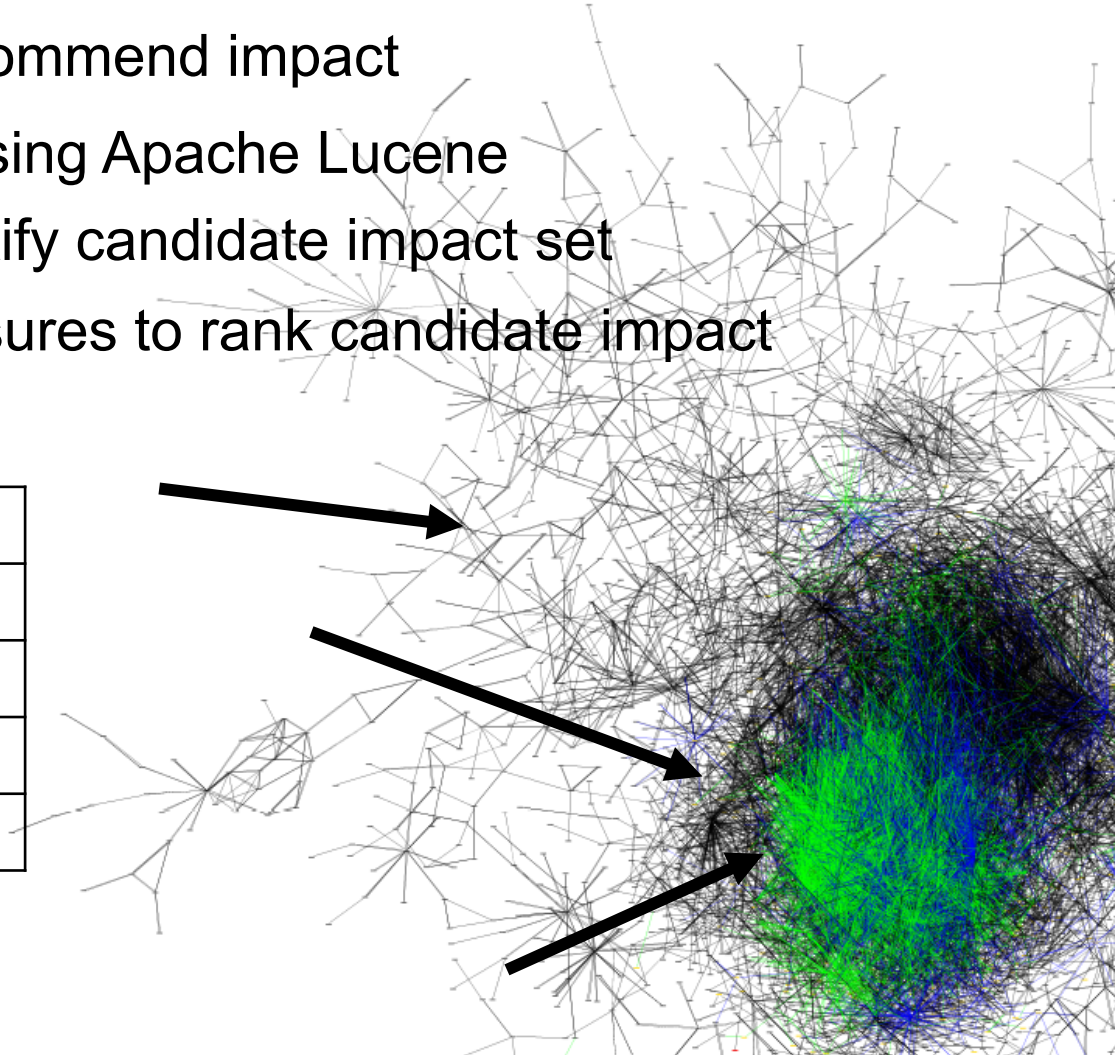
- Approach part 2: Recommend impact
  - Find similar bugs using Apache Lucene
  - Follow links to identify candidate impact set



Decision Support for Impact Analysis

- Approach part 2: Recommend impact
  - Find similar bugs using Apache Lucene
  - Follow links to identify candidate impact set
  - Use centrality measures to rank candidate impact

1. Requirement X.Y
2. Design Document X.Y
3. Test case UTC56
4. Design Document X.Y
5. Requirement Z.Y



CaseDigger 0.99

Input Tracker Case

Enter Tracker Case ID:

34533

bug in memory reporting tool

Search

WORKFLOW
1. Enter tracker case # to investigate
2. Type or paste textual description.
3. Click 'Search'
4. Analyze recommendations, give feedback.
5. Conclude feedback by pressing 'Done'

Conclude Feedback

Done

Most Similar Tracker Cases in the Knowledge Base

OK?	IA	Links	Sim.	ID	Title
☐			0,93	#45470	Sequence Test Framework bug
☑	X	2	0,74	#30946	Compiler Statistics tool in CB
☑			0,56	#23132	Reference handling bug in Shared Blob implementat
☑	X	9	0,54	#28889	Debug-tool, write-trap for all memory pools which are
☐	X	9	0,54	#30635	Debug-tool, write-trap for all memory pools which are
☐			0,53	#20550	Non-critical memory leaks in AC800M Connect
☐			0,52	#23558	Memory leakage during LEG session
☐			0,52	#42822	Minor bug in CVarAccessItem::Find
☐			0,49	#44345	Bug in MMS for EventNotification
☐			0,47	#34911	Error in Memory Leaks OPC Server
☐			0,47	#42821	Bug in the SFC editor
☐			0,45	#22294	Not possible to dump memory on compact flash in H
☐			0,45	#22223	Not possible to dump memory on compact flash in H
☐	X	2	0,45	#41624	Error in Illegal memory access handling of PM891
☐			0,44	#41867	Tool Routing Causes Large Heap Leakage and Cra
☐			0,42	#21526	GSD Import Tool has several errors in diagnostics cc
☐			0,42	#21357	GSD Import Tool has several errors in diagnostics cc

Reported Impact in Similar Tracker Cases

OK?	Conf.	ID	Type	Title
☐	1	pa-fct-340	Requirement	SYSTEM VE
☐	0,64	sr-dgn-015	Requirement	SIL3 / CAT4
☐	0,43	basichwlib	Hardware library	
☑	0,32	sr-fct-005	Requirement	APPLICATIO
☐	0,21	s900ioci854hwlib	Hardware library	
☐	0,19	3bse032067	Test description	FTTD MMU H
☑	0,18	3bse032066	Test description	DTD MMU H
☐	0,17	sr-fct-001	Requirement	
☐	0,17	mmu-dof-030	Requirement	
☑	0,17	mmu-dof-031	Requirement	
☐	0,17	mmu-dof-032	Requirement	
☐	0,17	mmu-dof-034	Requirement	
☐	0,16	mmu-dof-009	Requirement	
☐	0,15	sam-dof-410	Requirement	
☐	0,15	sam-dof-430	Requirement	
☐	0,14	sam-dof-420	Requirement	
☑	0,13	dip-dof-004	Requirement	BUILD PROJ
☑	0,13	3bse031917	Unspecified artifact	
☐	0,13	s900ioci851hwlib	Hardware library	
☐	0,13	3bse030861	Unspecified artifact	
☐	0,11	3bse055868	Test description	
☐	0,11	3bse048222	Test description	FTTD Access
☐	0,09	3bse051937	Test description	FTTD Module
☐	0,08	3bse032584	Test description	
☐	0,08	sr-fct-003	Requirement	

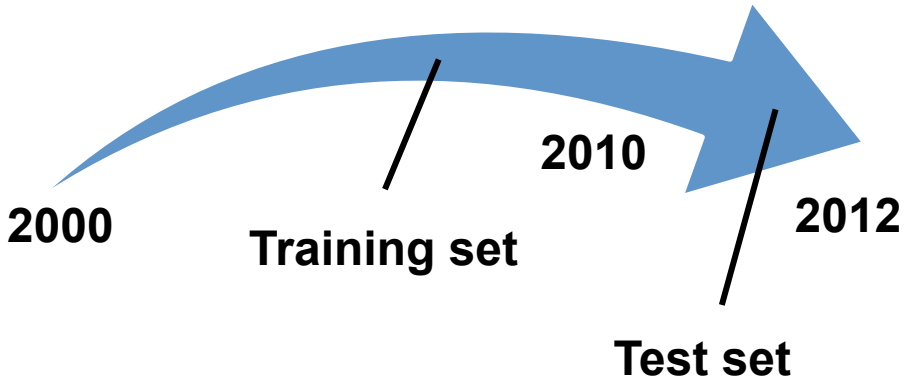
Tracker Case Details

Reference handling bug in Shared Blob implementation can cause crash in AC800
"We introduced a concept in SV4.0 called Shared Blob, which the aspect system developer can use to reduce the amount of memory needed when a lot of identical inherited aspects are activated at the same time. The unpacked blob is here only unplacked once and placed on the owning aspect. The inherited ASOs only have a reference to it. So far only AC800Connect (and our own Relative Name) has used it.We have now found a critical reference handling problem in this code. In some circumstances the DLL can be unloaded while there still are active shared blob objects. When the

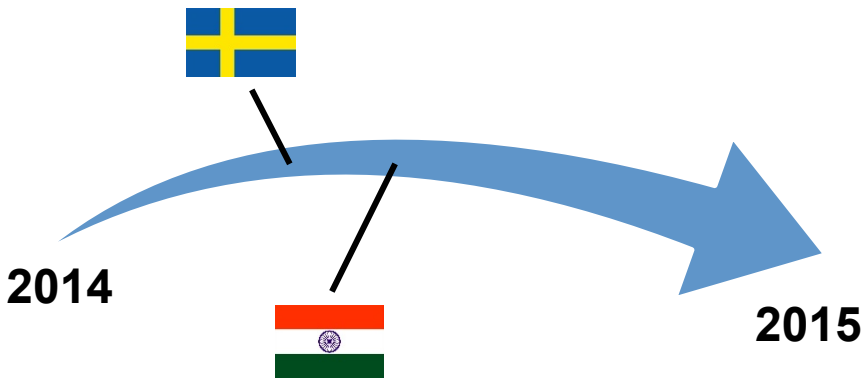

The Challenge
The Solution
The Evaluation

Evaluation Strategy

- Experiment: Replay historical inflow of issue reports

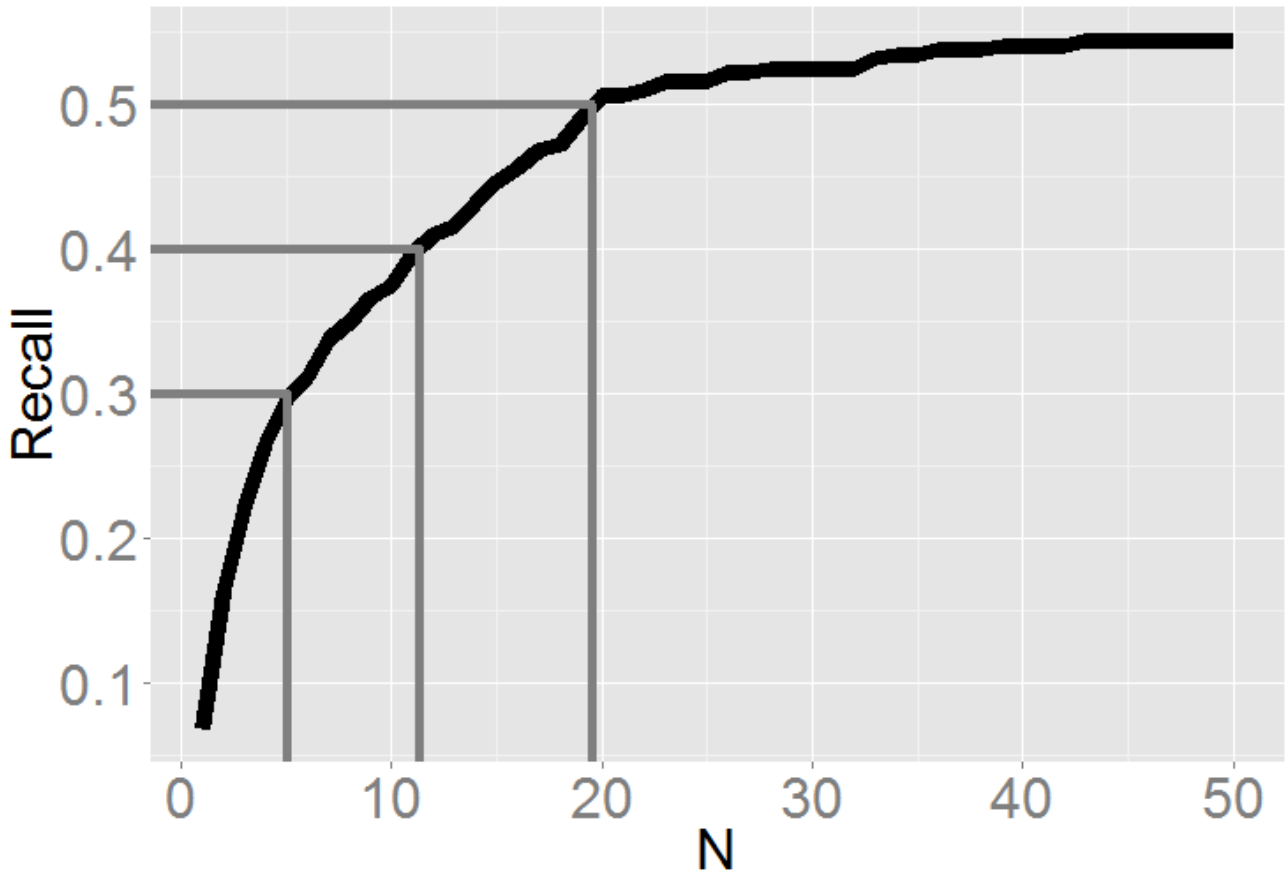


- Case study: Deploy ImpRec in two teams (11 developers)
  - Interviews
  - User log files



LUND UNIVERSITY

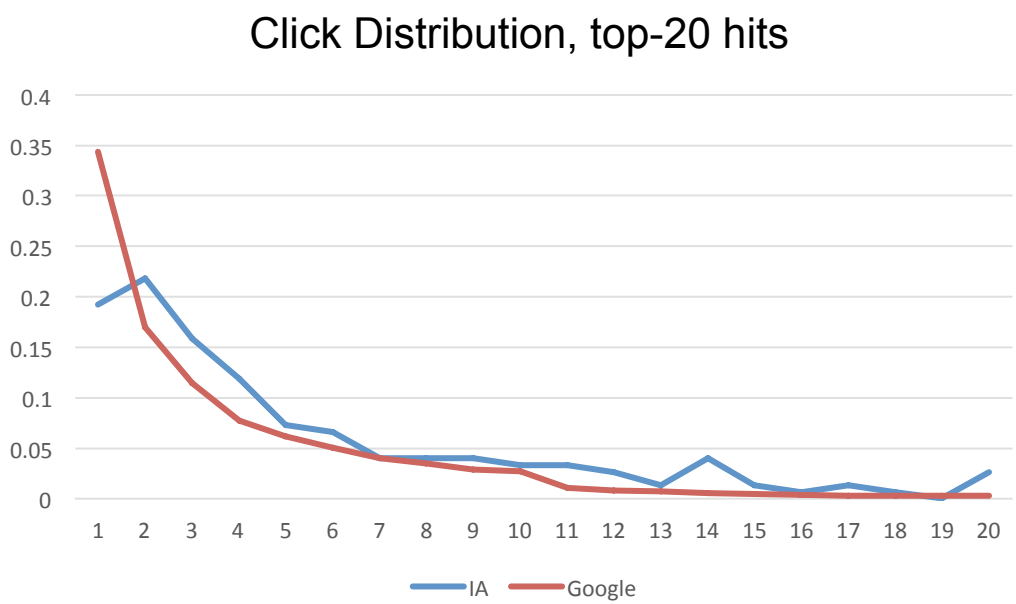
Experiment: Results



LUND UNIVERSITY

Case Study: Search Log Analysis

- Participants conducted 43 impact analyses
  - 70% of ImpRec uses provided relevant recommendations
  - Recall matches the experiment
  - Users missed 39% of the true recommendations



LUND UNIVERSITY

Case Study: Interviews

- Developers confirm ImpRec's potential
- Quick access to similar issues particularly well-received

"Finding these past bugs was exactly what I was looking for actually"

"The tool helped me to get a list of all related issues. The issue that I was working on was raised in many earlier system versions"

"I found it very useful as I was able to find some old issue reports with similar problems and how they were fixed"

Conclusions

LUND UNIVERSITY

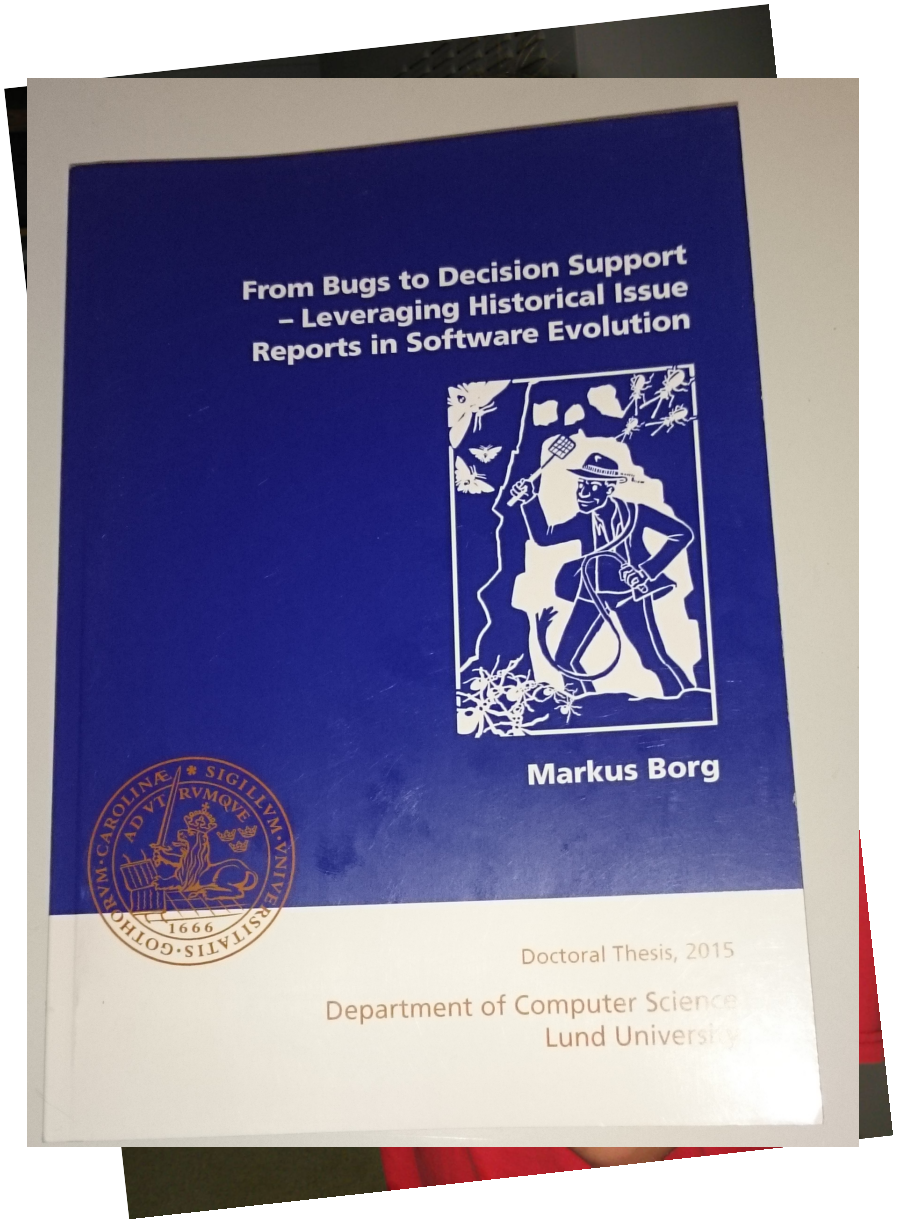
Decision Support for Impact Analysis

- Recommendation system provides a useful starting point
  - 30-50% recall enables jump-starting analysis
  - Provide warning if probable impact is missing
- Recommending related issues is a popular feature
  - Study previous issue resolutions
  - Compare with previous impact analyses

LUND UNIVERSITY

Open Questions

- How old project data can be used in the recommendation system?
  - Need to monitor the performance over time
- How to adapt current IA process when adding new decision support tool?
  - All new tools must be motivated in a formal report



Impact analysis is difficult, but old bugs are interesting!

- (Re)using traces from historical bug resolutions could be an approach to benefit more from traceability

LUND UNIVERSITY

Thank you!

markus.borg@cs.lth.se
cs.lth.se/markus_borg
@_Troddel_

PHOTO CREDITS

Brown stink bug
- Marlin E. Rice

Isopods
- Omoshiro Aquarium
- Flickr: littoraria, coda

Cubicles
- Flickr: templetonelliot, ifl, danburgmurmur

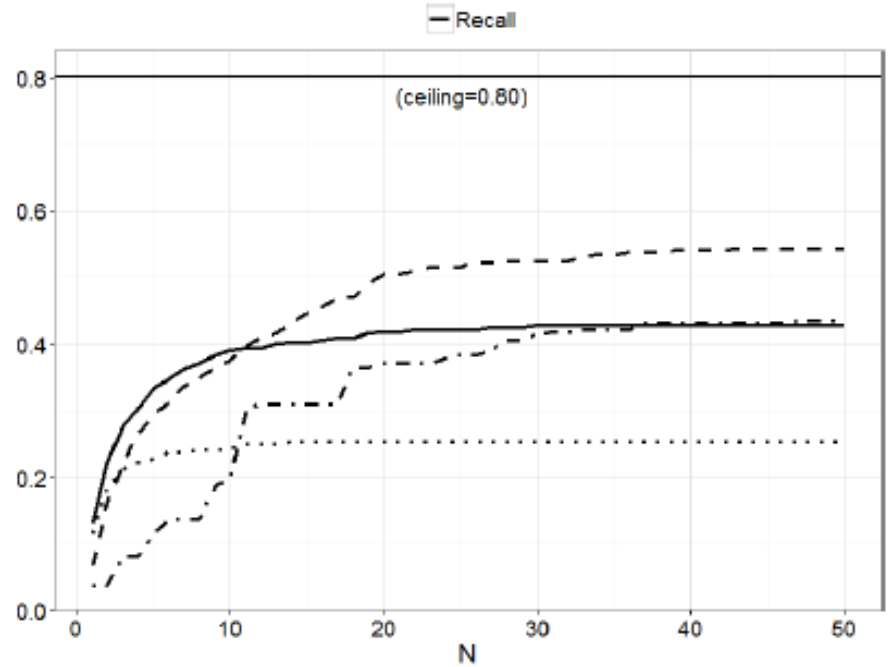
Eightball girl
- Flickr: mobilestreetlife

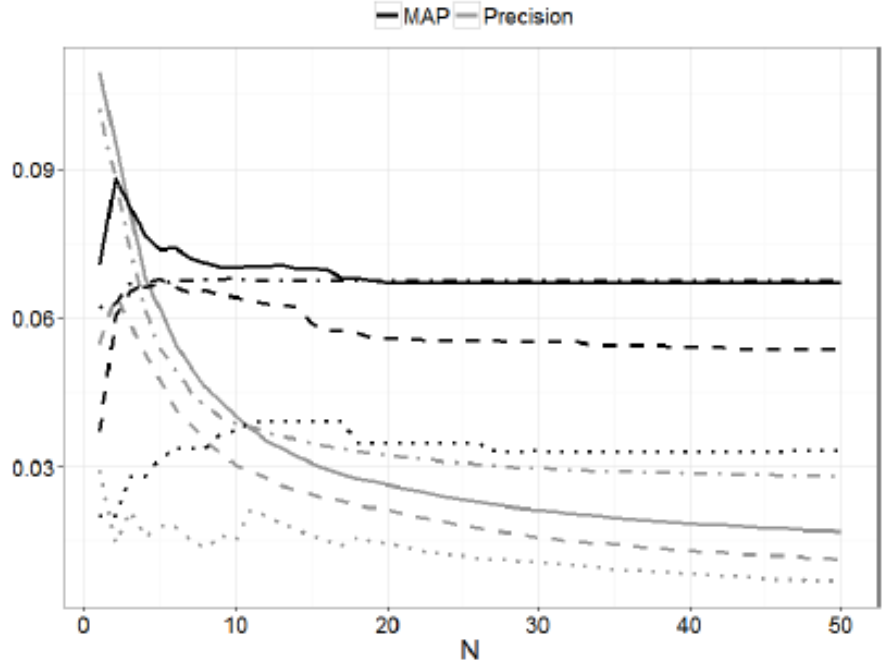
Evaluate
- Flickr: theideadesk

My wife
- My wife

LUND UNIVERSITY

Backup slides

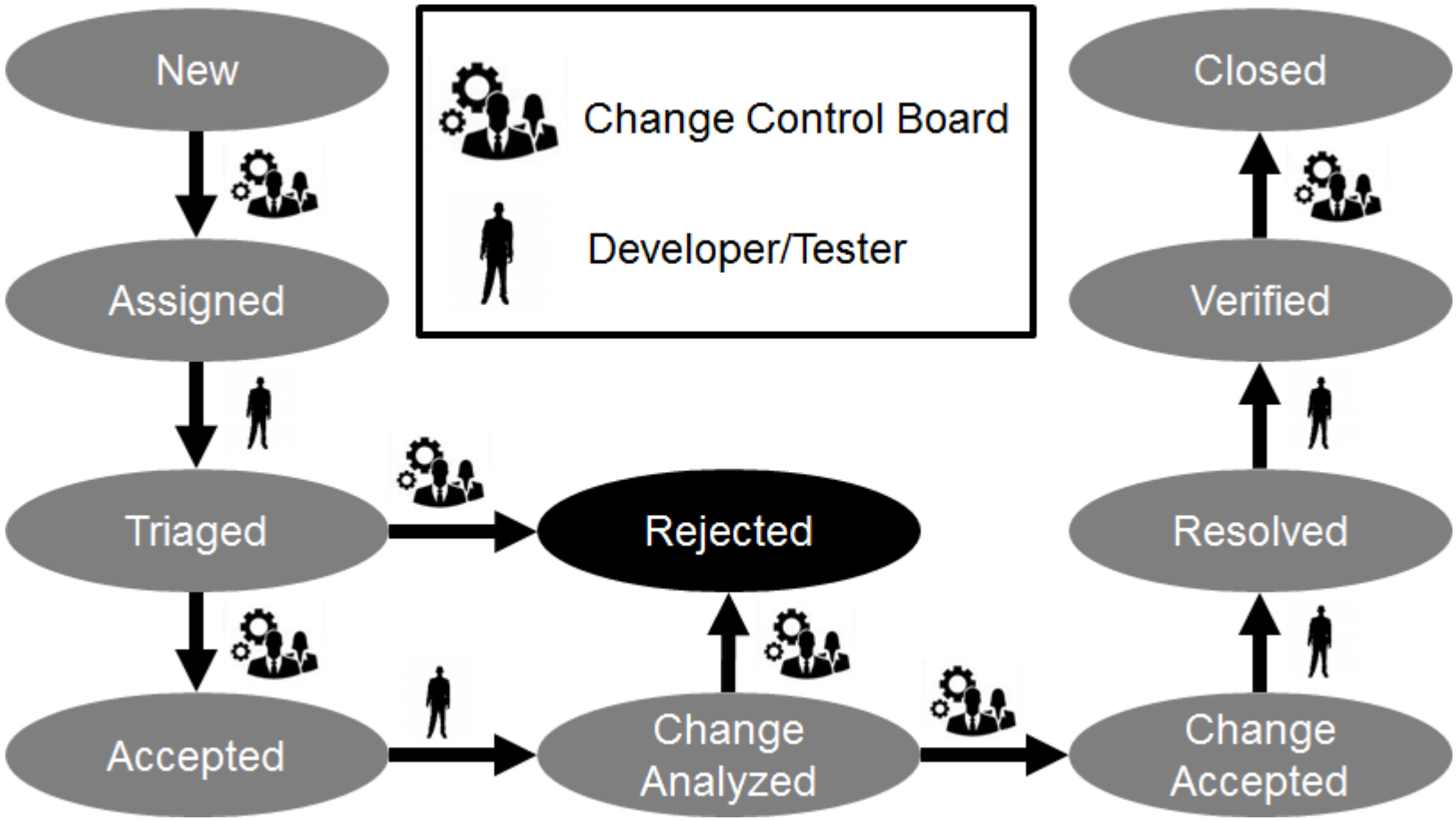




Solid line: ImpRec A, dashed line: ImpRec B, dot-dashed line: ImpRec Text, and dotted line: ImpRec Cent.

LUND UNIVERSITY

Change Management Process



Survey Details: Challenges

	N	Never	Few projects	Some projects	Most projects	Every project	*Median*
Insufficient tool support	90	11.1% (10)	21.1% (19)	17.8% (16)	**34.4% (31)**	15.6% (14)	*Most projects - Some projects*
Difficulty in estimating the effort required to manage a change	90	4.4% (4)	17.8% (16)	31.1% (28)	**36.7% (33)**	10% (9)	*Some projects*
Vast number of artefacts to trace	90	7.8% (7)	15.6% (14)	**35.6% (32)**	25.6% (23)	15.6% (14)	*Some projects*
Too coarse granularity of the traceability between artefacts to accurately know the consequences of a change	90	10% (9)	25.5% (23)	26.7% (24)	**28.9% (26)**	8.9% (8)	*Some projects*
Insufficient traceability between artefacts to accurately know the consequences of a change	90	5.6% (5)	25.6% (23)	**32.2% (29)**	28.9% (26)	7.8% (7)	*Some projects*
Difficulty in determining the effect of a change on system safety	90	4.4% (4)	23.3% (21)	**38.9% (35)**	24.4% (22)	8.9% (8)	*Some projects*
Long time for evaluating the consequences of a change	90	5.6% (5)	27.8% (25)	**34.4% (31)**	25.6% (23)	6.7% (6)	*Some projects*
Difficulty in assessing system-level impact of component reuse	90	10% (9)	28.9% (26)	**38.9% (35)**	18.9% (17)	3.3% (3)	*Some projects*
Unclear meaning of the traceability between artefacts in order to know how to manage a change	90	15.6% (14)	26.7% (24)	**35.6% (32)**	17.8% (16)	4.4% (4)	*Some projects*
Insufficient confidence by assessor or certifiers in having managed a change properly	90	20% (18)	27.8% (25)	**32.2% (29)**	16.7% (15)	3.3% (3)	*Some projects*
Lack of a systematic process for performing impact analysis	90	12.2% (11)	**28.9% (26)**	27.8% (25)	20% (18)	11.1% (10)	*Some projects*
Difficulty in deciding if a component can be reused	90	21.1% (19)	30% (27)	**31.1% (28)**	14.4% (13)	3.3% (3)	*Few projects*
Excessive detail of the traceability between artefacts, making traceability management more complex than necessary for impact analysis purposes	90	25.6% (23)	**32.2% (29)**	26.7% (24)	11.1% (10)	4.4% (4)	*Few projects*

LUND UNIVERSITY

Survey Details: Levels of Automation

Table 7. Level of automation offered by tools for SECIA from each artefact type

	N	Fully Manual	Decision Support Available	Semi-Automated Recommendations	Highly-Automated Recommendations	Automatic Impact Analysis	***Median***
Source Code	73	**31.5% (23)**	16.4% (12)	**31.5% (23)**	17.8% (13)	2.8% (2)	*Semi-Automated Recommendations*
Traceability Specifications	79	25.3% (20)	26.6% (21)	**27.8% (22)**	15.2% (12)	5.1% (4)	*Decision Support Available*
Architecture Specifications	72	34.7% (25)	**41.7% (30)**	19.4% (14)	1.4% (1)	2.8% (2)	*Decision Support Available*
Tool-Supported V&V Results	79	**32.9% (26)**	21.5% (17)	24.1% (19)	17.7% (14)	3.8% (3)	*Decision Support Available*
Test Case Specifications	79	**39.2% (31)**	29.1% (23)	20.3% (16)	8.9% (7)	2.5% (2)	*Decision Support Available*
Requirements Specifications	80	**40% (32)**	33.8% (27)	16.2% (13)	8.7% (7)	1.3% (1)	*Decision Support Available*
Safety Analysis Results	76	**40.8% (31)**	23.7% (18)	23.7% (18)	10.5% (8)	1.3% (1)	*Decision Support Available*
Design Specifications	76	**42.1% (32)**	35.5% (27)	17.1% (13)	4% (3)	1.3% (1)	*Decision Support Available*
Safety Cases	73	**56.1% (41)**	27.4% (20)	13.7% (10)	1.4% (1)	1.4% (1)	*Fully Manual*
Manual V&V Results	78	**56.4% (44)**	23.1% (18)	16.7% (13)	3.8% (3)	0% (0)	*Fully Manual*
Reused Components Information	71	**59.2% (42)**	31% (22)	7% (5)	1.4% (1)	1.4% (1)	*Fully Manual*
Personnel Competence Specifications	66	**63.6% (42)**	28.8% (19)	7.6% (5)	0% (0)	0% (0)	*Fully Manual*
System Lifecycle Plans	75	**65.4% (49)**	21.3% (16)	9.3% (7)	4% (3)	0% (0)	*Fully Manual*
Assumptions and Operation Conditions Specifications	72	**68.1% (49)**	20.8% (15)	9.7% (7)	1.4% (1)	0% (0)	*Fully Manual*

Changes to what Artifacts Trigger IA?

1. Requirements Specifications
2. Source Code
3. Test Case Specifications

Table 2. SECIA frequency as a consequence of changes in artefact types

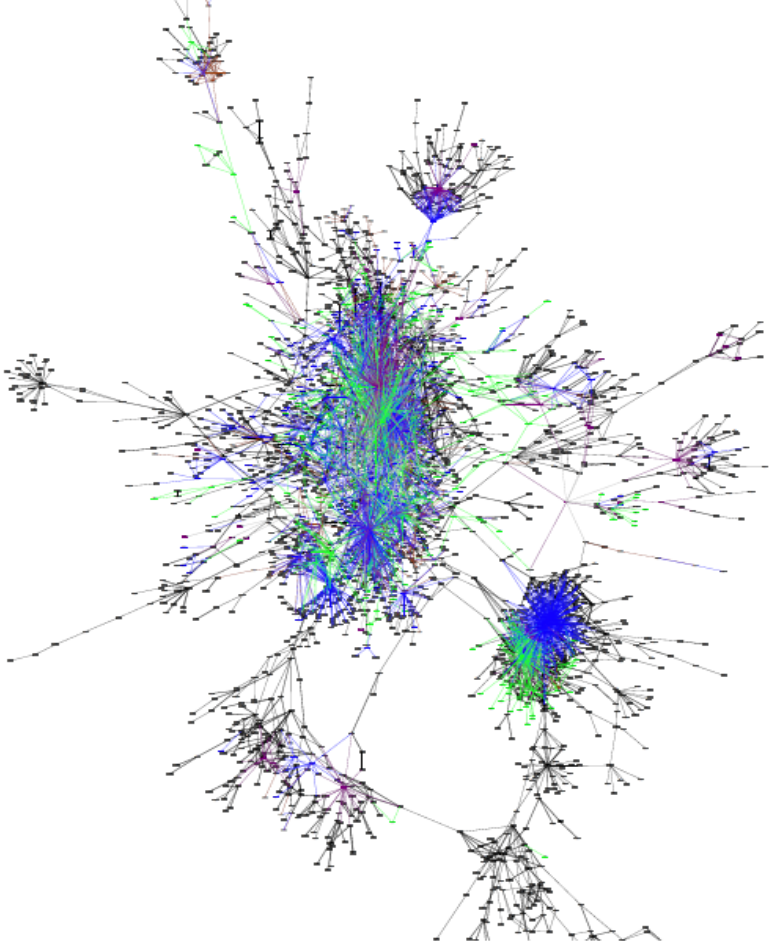
	N	Never	Few projects	Some projects	Most projects	Every project	*Median*
Requirements Specifications	78	3.8% (3)	9% (7)	25.6% (20)	23.1% (18)	**38.5% (30)**	*Most projects*
Source Code	74	13.5% (10)	16.2% (12)	16.2% (12)	20.3% (15)	**33.8% (25)**	*Most projects*
Test Case Specifications	77	9.1% (7)	16.9% (13)	22.1% (17)	20.8% (16)	**31.1% (24)**	*Most projects*
Traceability Specifications	78	10.3% (8)	21.8% (17)	12.8% (10)	24.3% (19)	**30.8% (24)**	*Most projects*
Design Specifications	76	7.9% (6)	13.1% (10)	25% (19)	23.7% (18)	**30.3% (23)**	*Most projects*
Safety Analysis Results	76	3.9% (3)	22.4% (17)	19.7% (15)	26.3% (20)	**27.7% (21)**	*Most projects*
Manual V&V Results	76	9.2% (7)	23.7% (18)	**26.3% (20)**	14.5% (11)	**26.3% (20)**	*Some projects*
Safety Cases	77	10.4% (8)	22.1% (17)	**27.2% (21)**	14.3% (11)	26% (20)	*Some projects*
Assumptions and Operation Conditions Specs.	73	11% (8)	20.5% (15)	**32.9% (24)**	16.4% (12)	19.2% (14)	*Some projects*
Tool-Supported V&V Results	76	18.4% (14)	22.4% (17)	**25% (19)**	13.2% (10)	21% (16)	*Some projects*
Architecture Specifications	71	**22.6% (16)**	21.1% (15)	18.3% (13)	19.7% (14)	18.3% (13)	*Some projects*
System Lifecycle Plans	76	23.7% (18)	**25% (19)**	18.4% (14)	15.8% (12)	17.1% (13)	*Some projects*
Reused Components Information	72	20.8% (15)	**29.5% (21)**	16.7% (12)	18% (13)	15.3% (11)	*Few projects - Some projects*
Personnel Competence Specifications	70	**40% (28)**	24.3% (17)	14.3% (10)	8.6% (6)	12.8% (9)	*Few projects*

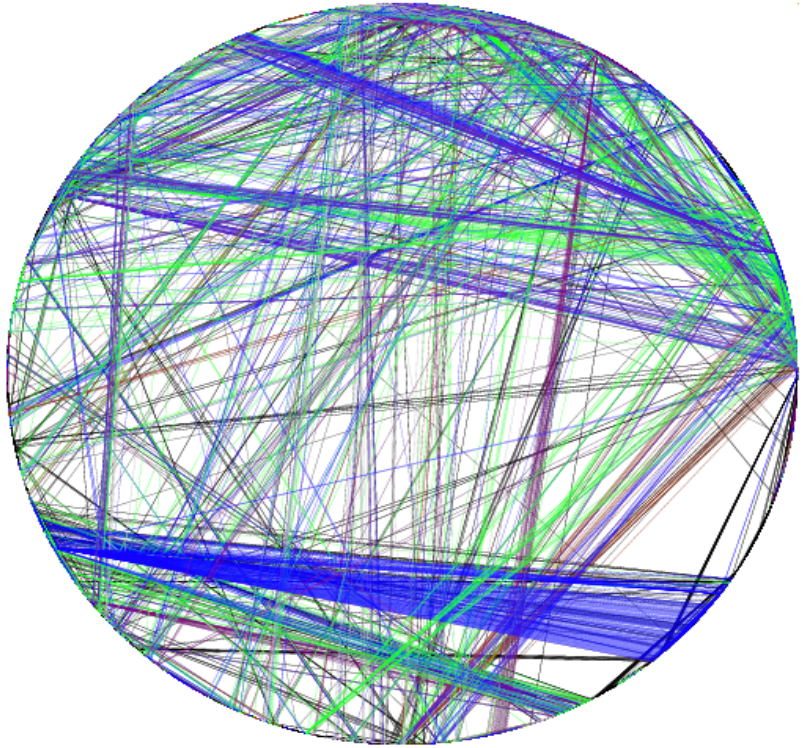
What Artifacts are Reported as Impacted?

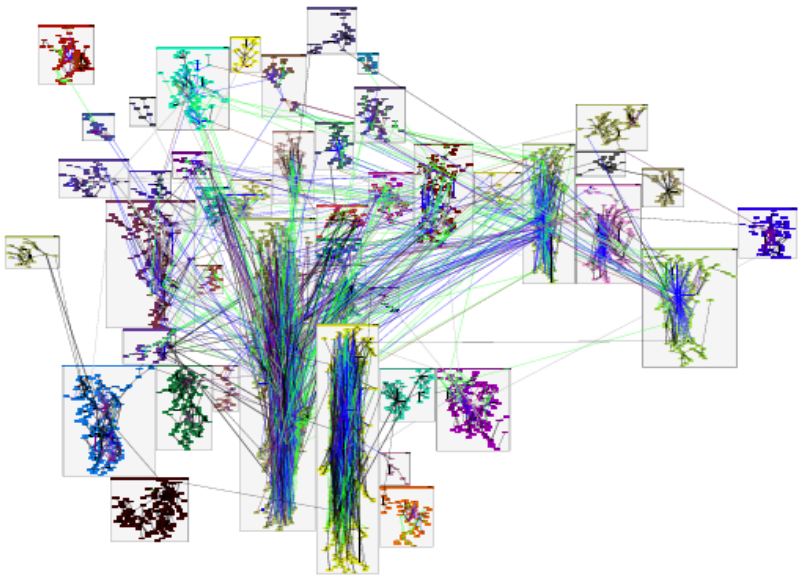
1. Manual V&V Results
2. Test Case Specifications
3. Source Code

Table 4. Change impact frequency on artefact types

	N	Never	Few projects	Some projects	Most projects	Every project	*Median*
Manual V&V Results	74	4.1% (3)	18.9% (14)	25.7% (19)	24.3% (18)	**27% (20)**	*Most projects*
Test Case Specifications	77	3.9% (3)	15.6% (12)	**31.1% (24)**	27.3% (21)	22.1% (17)	*Some projects*
Source Code	74	2.7% (2)	14.9% (11)	**33.8% (25)**	21.6% (16)	27% (20)	*Some projects*
Safety Cases	73	6.9% (5)	21.9% (16)	23.3% (17)	21.9% (16)	**26% (19)**	*Some projects*
Requirements Specifications	76	5.3% (4)	18.4% (14)	**31.6% (24)**	15.8% (12)	28.9% (22)	*Some projects*
Safety Analysis Results	73	4.1% (3)	23.3% (17)	**30.1% (22)**	17.8% (13)	24.7% (18)	*Some projects*
Design Specifications	76	1.3% (1)	25% (19)	**32.9% (25)**	17.1% (13)	23.7% (18)	*Some projects*
Traceability Specifications	74	10.8% (8)	24.3% (18)	**25.7% (19)**	14.9% (11)	24.3% (18)	*Some projects*
Architecture Specifications	75	10.7% (8)	25.3% (19)	**37.3% (28)**	10.7% (8)	16% (12)	*Some projects*
Assumptions and Operation Conditions Specs.	71	14.1% (10)	**29.6% (21)**	26.7% (19)	12.7% (9)	16.9% (12)	*Some projects*
Tool-Supported V&V Results	73	13.7% (10)	**37% (27)**	17.8% (13)	13.7% (10)	17.8% (13)	*Few projects*
System Lifecycle Plans	75	22.7% (17)	**29.3% (22)**	22.7% (17)	10.7% (8)	14.6% (11)	*Few projects*
Reused Components Information	70	21.4% (15)	**31.4% (22)**	25.7% (18)	11.5% (8)	10% (7)	*Few projects*
Personnel Competence Specifications	68	**39.7% (27)**	30.9% (21)	16.2% (11)	7.3% (5)	5.9% (4)	*Few projects*







LUND
UNIVERSITY

LUND
UNIVERSITY