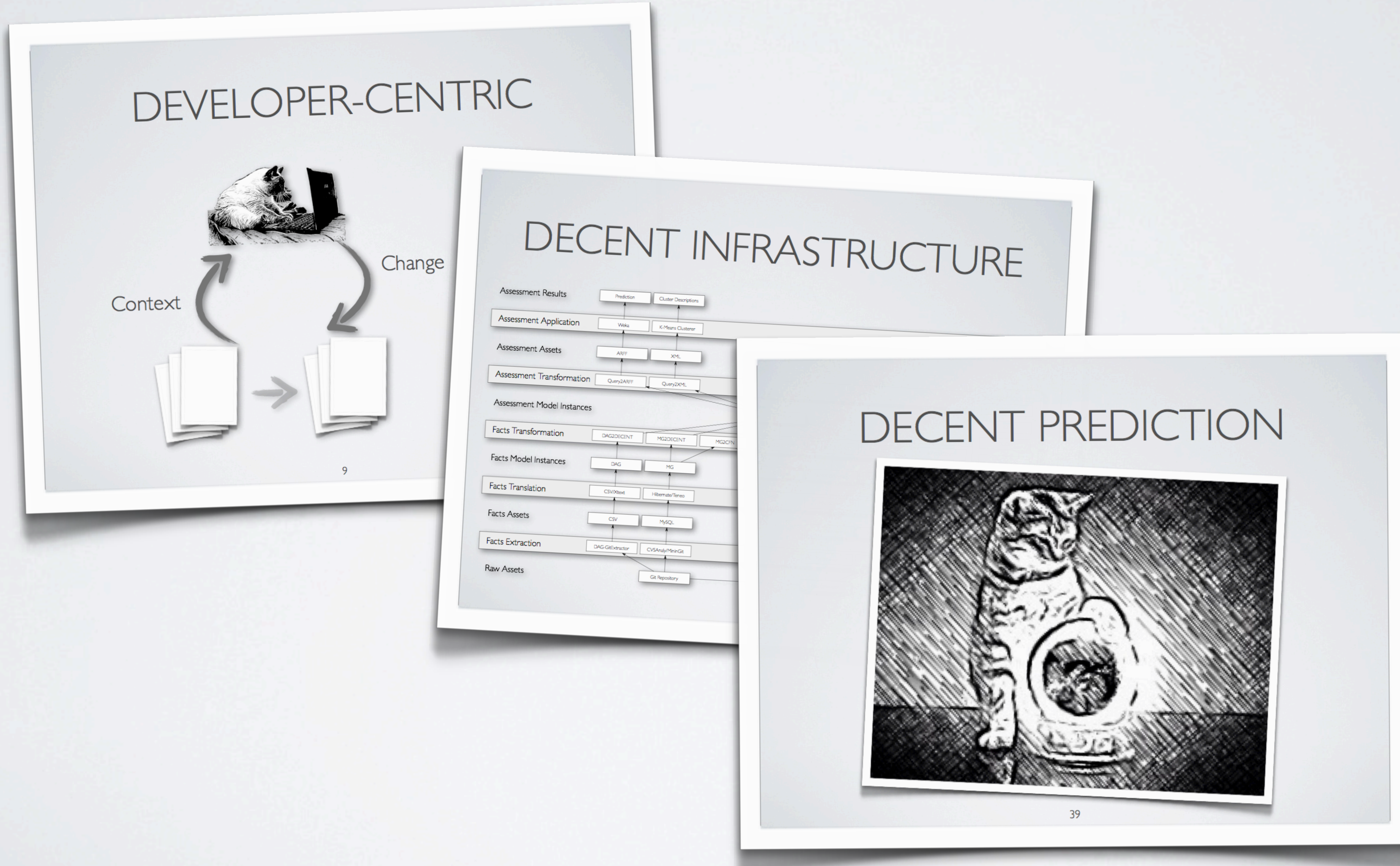
Software Engineering for Distributed Systems
Georg-August-University of Göttingen

DEVELOPER-CENTRIC SOFTWARE ASSESSMENT

Philip Makedonski
Jens Grabowski

OVERVIEW



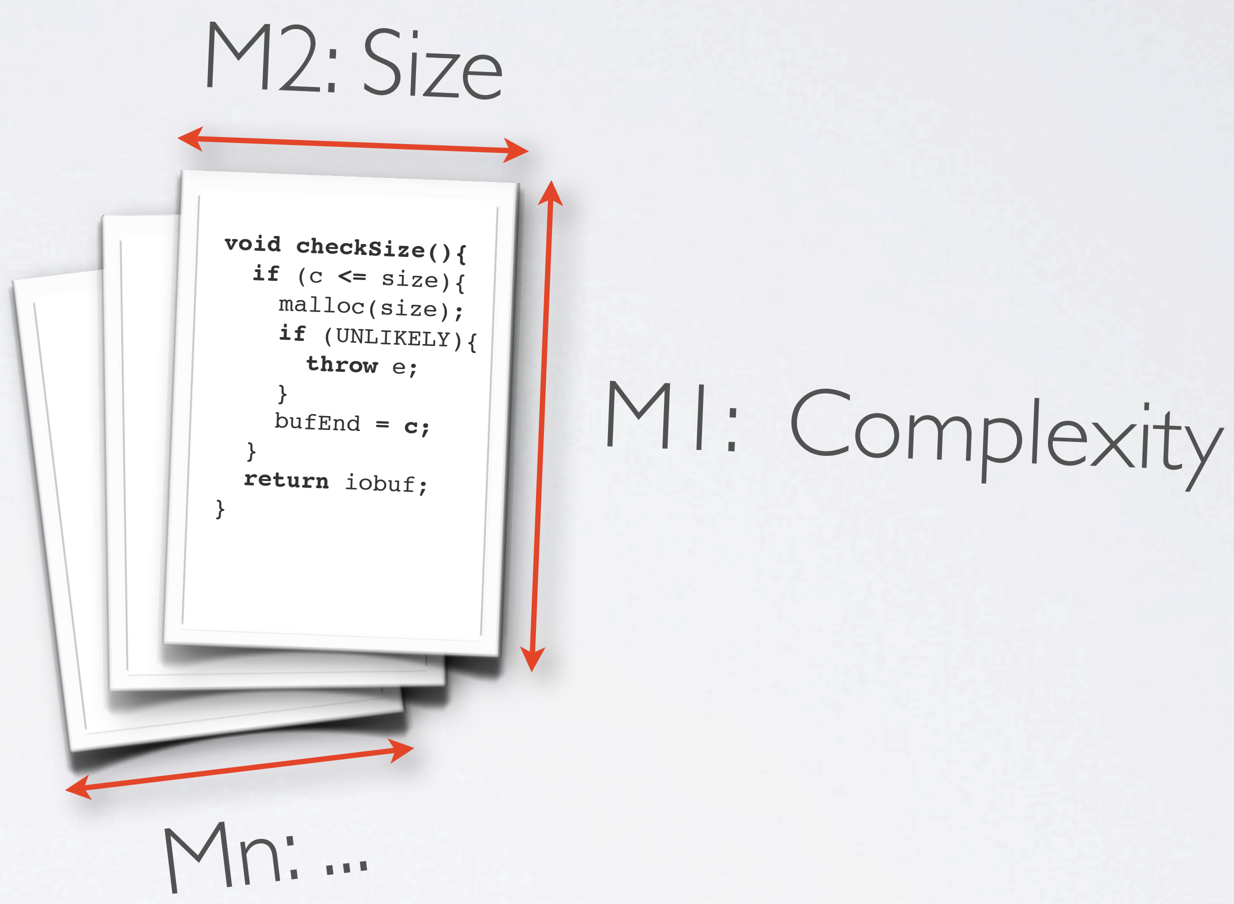
DEVELOPER-CENTRIC
Context
Change
9
DECENT INFRASTRUCTURE
Assessment Results
Prediction
Cluster Descriptions
Assessment Application
Weka
K-Means Clusterer
Assessment Assets
ARFF
XML
Assessment Transformation
Query2ARFF
Query2XML
Assessment Model Instances
Facts Transformation
DAG2DECENT
MG2DECENT
MG2CFN
Facts Model Instances
DAG
MG
Facts Translation
CSV/Xtext
Hibernate/Teneo
Facts Assets
CSV
MySQL
Facts Extraction
DAG-GitExtractor
CVSAnaly/MininGit
Raw Assets
Git Repository
DECENT PREDICTION
39

SOFTWARE ASSESSMENT

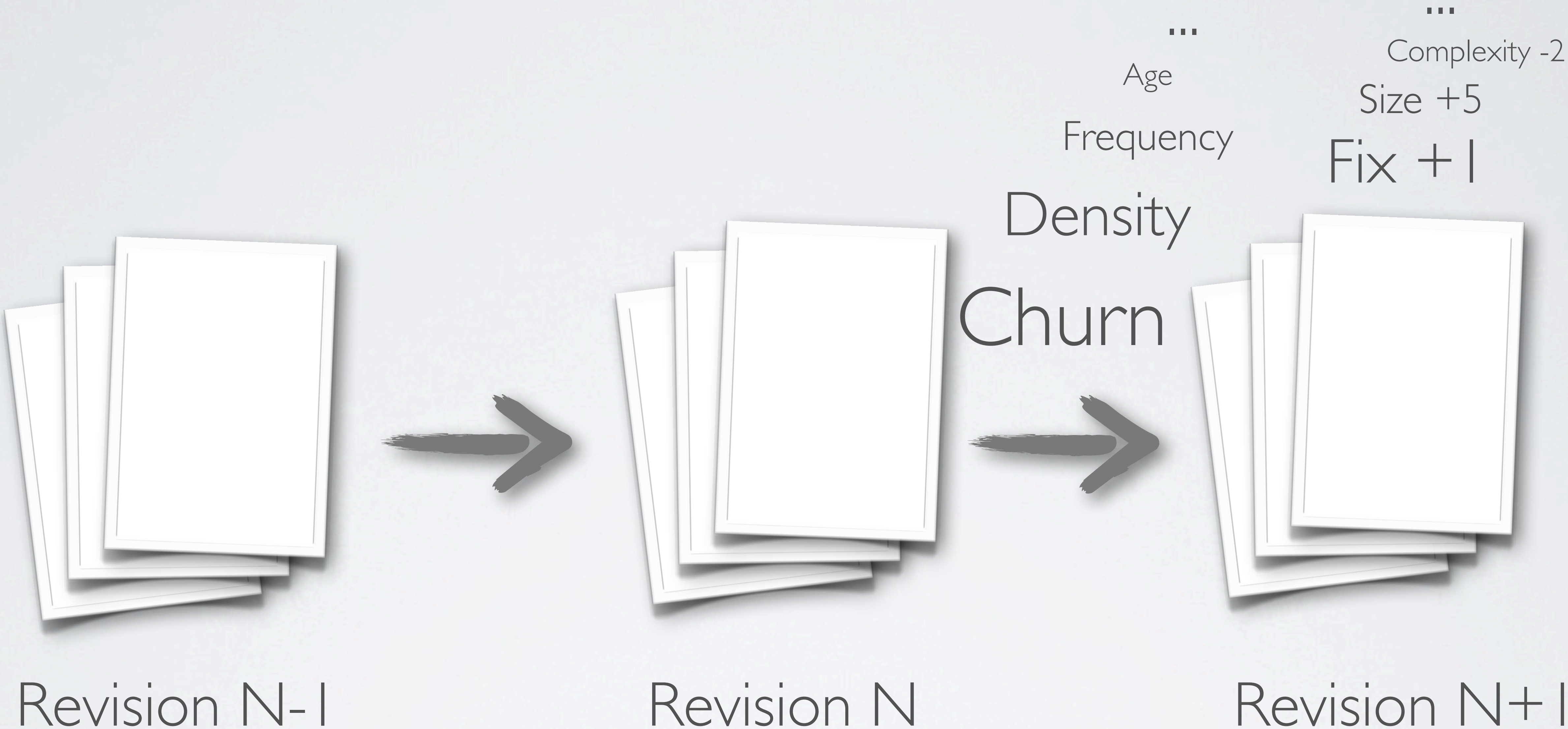
- Software Assessment
  - *"the process of posing specific questions about the software system under study and carrying out specialized analyses to answer these questions"* (Nierstrasz, 2012)
- Agile Software Assessment
  - *"a meta-tooling infrastructure and environment that allows rapid and cheap development of custom lightweight tools to support software assessment and program understanding"* (Nierstrasz, 2012)

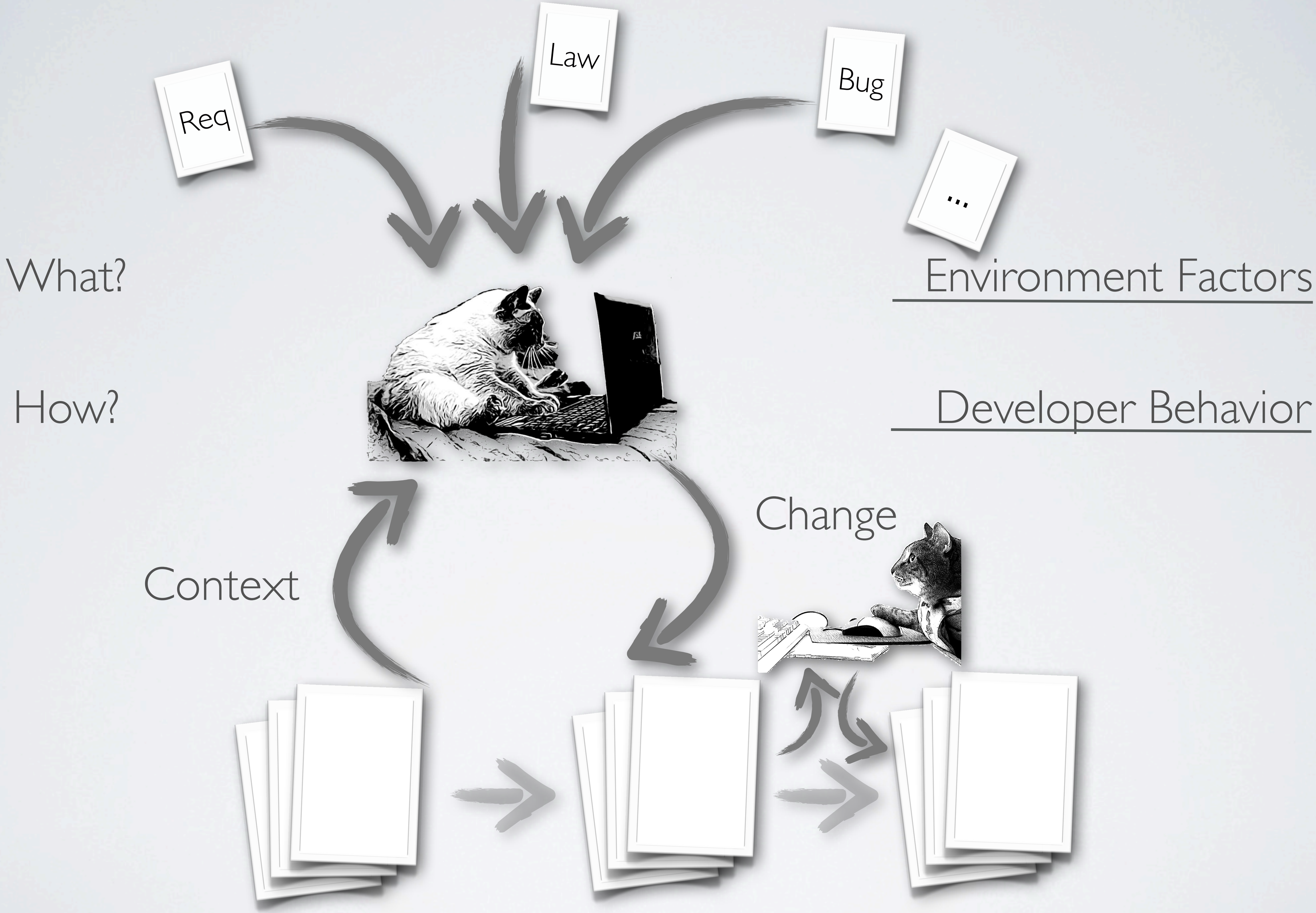
ARTIFACT-CENTRIC

- Metrics
- Clones
- Dependencies
- Domain
- ...



CHANGE-CENTRIC




Req
Law
Bug
...
What?
Environment Factors
How?
Developer Behavior
Change
Context

DEVELOPER-CENTRIC

- Behavior
- Experience
- Interactions
- Ownership
- ...

DECENT

Developer-Centric Software Assessment

An Industrial Study on the Risk of Software Changes

Emad Shihab and Ahmed E. Hassan
Software Analysis and Intelligence Lab (SAIL)
Queen's University, Canada
{emads, ahmed}@cs.queensu.ca

Bram Adams
Lab on Maintenance, Construction and Intelligence of Software (MCIS)
École Polytechnique de Montréal, Canada
bram.adams@polymtl.ca

Zhen Ming Jiang
Research In Motion
Waterloo, ON, Canada

ABSTRACT

Modelling and understanding bugs has been the focus of much of the Software Engineering research today. However, organizations are interested in more than just bugs. In particular, they are more concerned about managing risk, i.e., the likelihood that a code or design change will cause a negative impact on their products and processes, regardless of whether or not it introduces a bug. In this paper, we conduct a year-long study involving more than 450 developers of a large enterprise, spanning more than 60 teams, to better understand risky changes, i.e., *changes for which developers believe that additional attention is needed in the form of careful code or design reviewing and/or more testing*. Our findings show that different developers and different teams have their own criteria for determining risky changes. Using factors extracted from the changes and the history of the files modified by the changes, we are able to accurately identify risky changes with a recall of more than 67%, and a precision improvement of 87% (using developer specific models) and 37% (using team specific models), over a random model. We find that the number of lines and chunks of code added by the change, the bugginess of the files being changed, the number of bug reports linked to a change and the developer experience are the best indicators of change risk. In addition, we find that when a change has many related changes, the reliability of developers in marking risky changes is negatively affected. Our findings and models are being used today in practice to manage the risk of software projects.

Categories and Subject Descriptors

a survey of 600 firms showed that 35% of them had at least one runaway project [6]. Another study showed that, industry-wide, only 16.2% of software projects are on time and on budget. Of the rest, 52.7% are delivered with reduced functionality and 31.1% are cancelled before completion. The main reason for this large amount of late projects is the lack of proper software risk management (i.e., activities used to manage the possibility of harm or loss) [6, 10].

Due to the importance of risk management in the success of software projects, researchers and industry have become more interested and active in the area of software risk management [13, 23]. One line of work that received a large amount of attention recently is software bug prediction, where code and/or historical metrics are used to predict where bugs might appear in the future (e.g., [26, 35]). In fact a recent literature review showed that in the past ten years more than 200 papers were published on defect prediction alone [17].

However, organizations are interested in effective management of risk in general, which covers more than just bugs. For example, a recent initiative on managing technical debt aims at studying how compromises that developers make today will affect their software in the future [30]. Risky changes could introduce bugs but they could also delay the release of projects, and/or negatively impact customer satisfaction. For example, changes that might have a widespread impact on the code (e.g., switching threading models) or on the user (e.g., making the software application autosave every 1 min instead of 30 seconds, for optimization reasons) are considered risky, regardless of whether or not they introduce bugs. The risk is caused by the uncertainty introduced by the changes.

An Industrial Study on the Risk of Software Changes

Emad Shihab and Ahmed E. Hassan
Software Analysis and Intelligence Lab (SAIL)
Queen's University, Canada
{emads, ahmed}@cs.queensu.ca

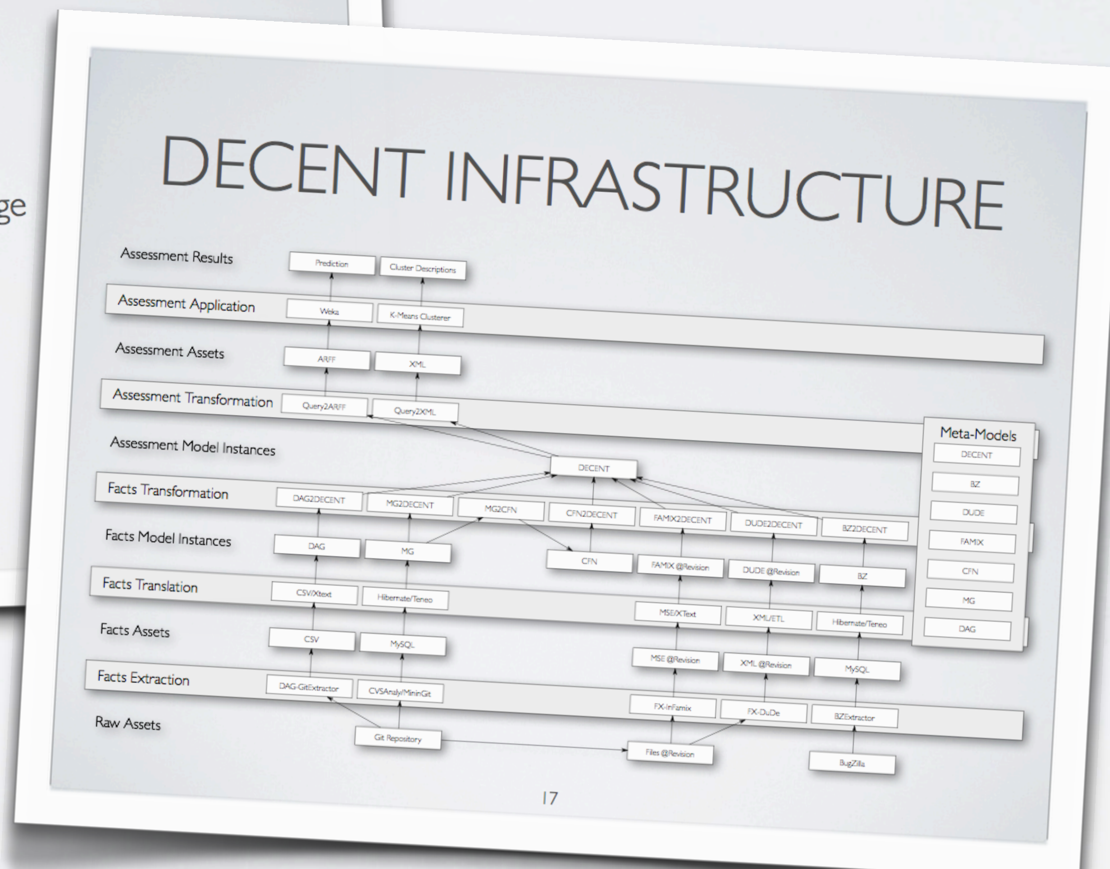
Bram Adams
Lab on Maintenance, Construction and Intelligence of Software (MCIS)
École Polytechnique de Montréal, Canada
bram.adams@polymtl.ca

Zhen Ming Jiang
Research In Motion
Waterloo, ON, Canada

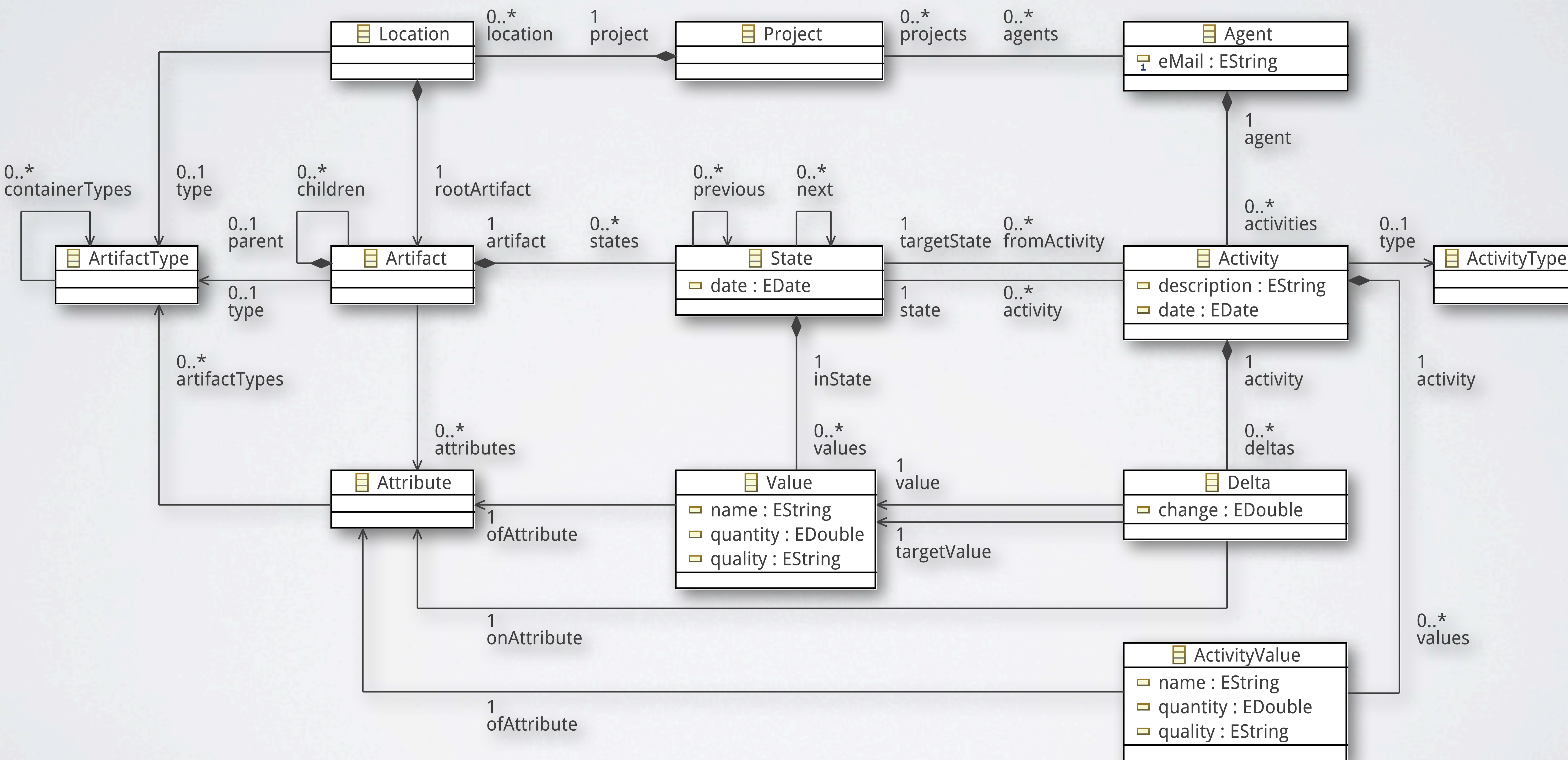
- Year-long study with 450+ developers from 60+ teams at RIM
- Focus on **risky** rather than **buggy** changes
- Different developers and teams have their own criteria
- 23 factors across 6 dimensions

OVERVIEW


DEVELOPER-CENTRIC
Context
Change
9


DECENT INFRASTRUCTURE
Assessment Results
Assessment Application
Assessment Assets
Assessment Transformation
Assessment Model Instances
Facts Transformation
Facts Model Instances
Facts Translation
Facts Assets
Facts Extraction
Raw Assets
Meta-Models
17

DECENT META-MODEL



Assessment Results

Assessment Application

Assessment Assets

Assessment Transformation

Assessment Model Instances DECENT

Facts Transformation

Facts Model Instances

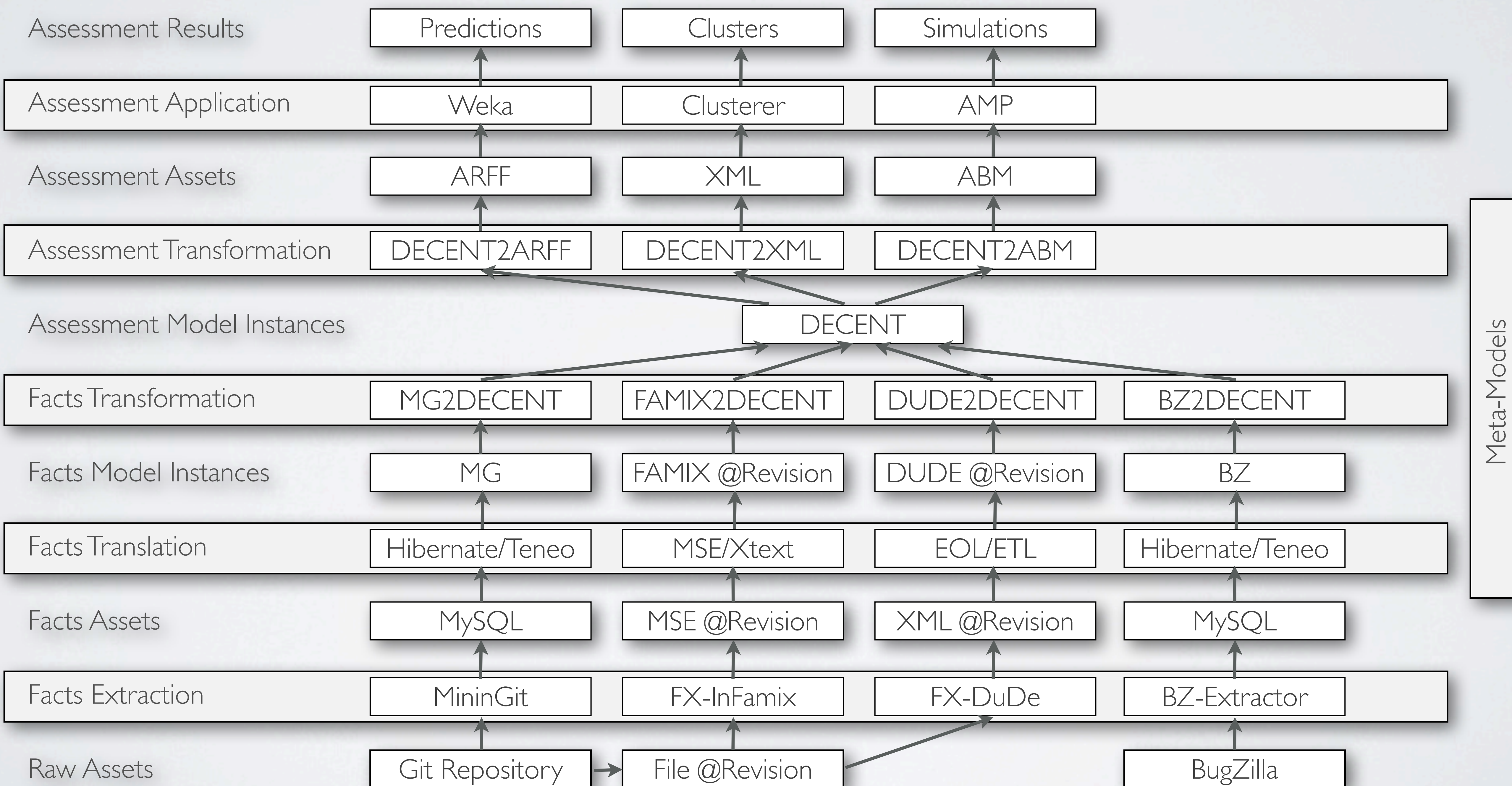
Facts Translation

Facts Assets

Facts Extraction

Raw Assets

DECENT INFRASTRUCTURE



2011 27th IEEE International Conference on Software Maintenance (ICSM)

Evidence-based Software Process Recovery: A Post-doctoral View

Abram Hindle

Department of Computer Science
University of California, Davis
Davis, CA
ah@softwareprocess.es

***Abstract*—Software development processes are often viewed as a panacea for software quality: prescribe a process and a quality project will emerge. Unfortunately this has not been the case, as practitioners are prone to push against processes that they do not perceive as helpful, often much to the dismay of stakeholders such as their managers. Yet practitioners still tend to follow some sort of software development processes regardless of the prescribed processes. Thus if a team wants to recover the software development processes of a project or if team is trying to achieve a certification such as ISO9000 or CMM, the team will be tasked with describing their development processes. Previous research has tended to focus on modifying existing projects in order to extract process related information. In contrast, our approach of *software process recovery* attempts to analyze software artifacts extracted from software repositories in order to infer the underlying software development processes visible within these software repositories.**

I. Introduction

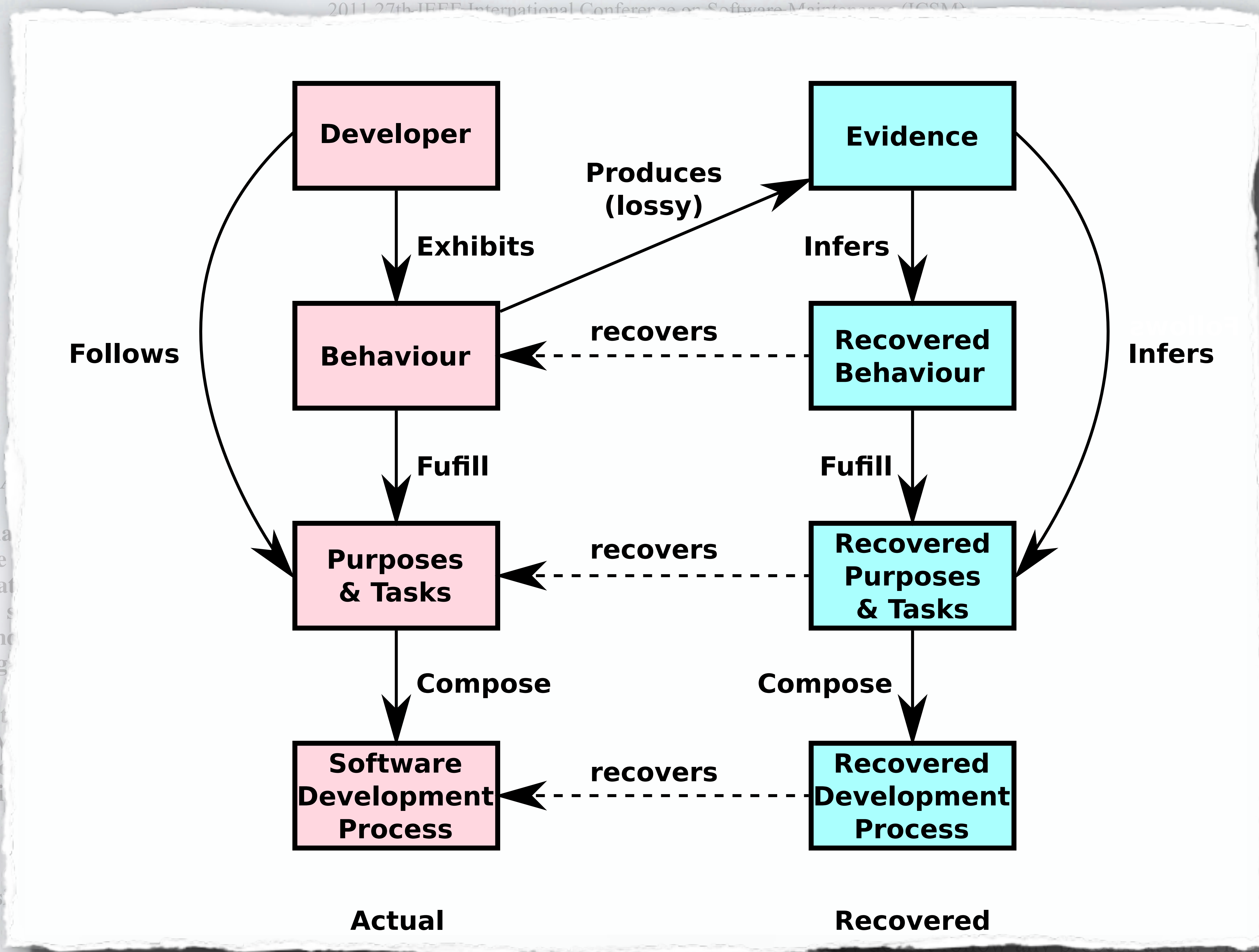
If one approaches a developer and asks them what software development process are they following, how will they answer? Will they respond with the process that their man-

extraction and validation of software processes being followed in practice, based on information extracted from the software repositories utilized by developers.

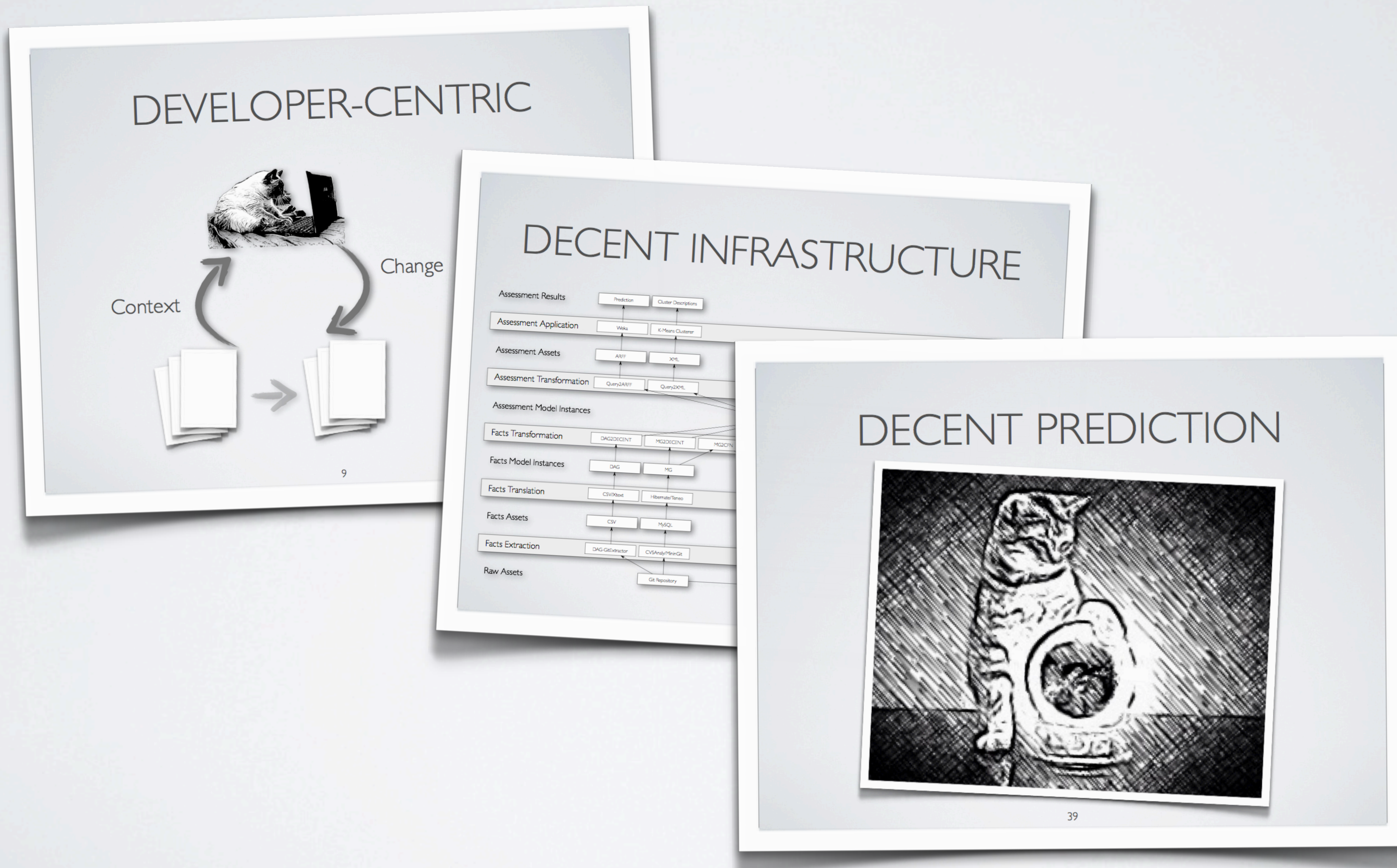
A. Stakeholder motivations

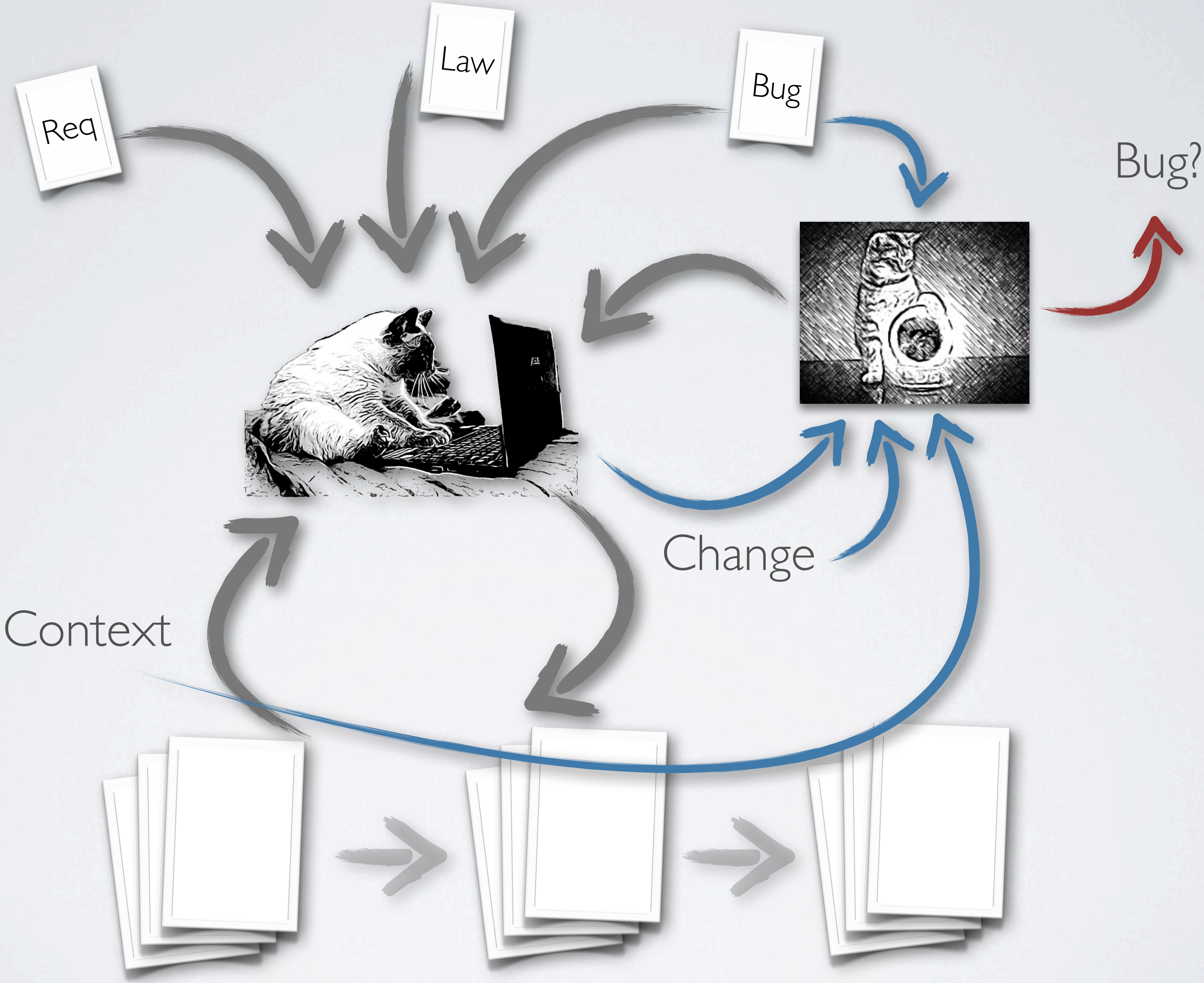
Recovering software development processes from existing projects is useful to many stakeholders who care about the system and also have some stake in the processes that govern its development.

Developers care about process in the sense that they are forced to follow it but also at the same time are forced to rely upon it. If developers act inconsistently, they create confusion based on the assumptions that other developers are making about development. Developers are surprised by behaviour that does not fit within an accepted process. Many developers would assume they do not follow any process at all. This is not the case as many developers, we would claim, follow a natural process based on routines they like to follow. These actions might result in greater software quality and thus motivate these

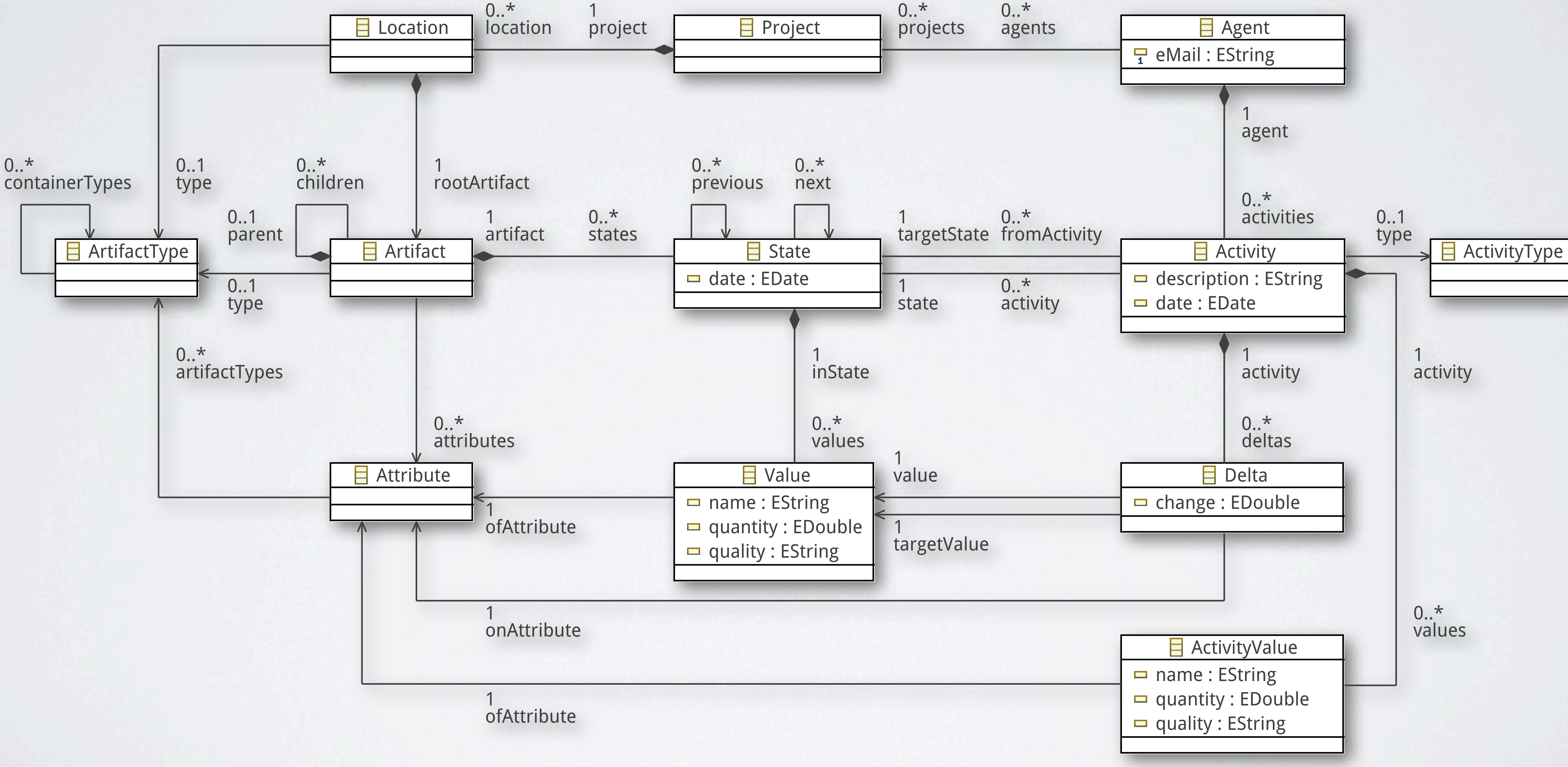

Developer
Evidence
Produces (lossy)
Exhibits
Infers
Follows
Behaviour
recovers
Recovered Behaviour
Infers
Fufill
Fufill
Purposes & Tasks
recovers
Recovered Purposes & Tasks
Compose
Compose
Software Development Process
recovers
Recovered Development Process
Actual
Recovered

OVERVIEW

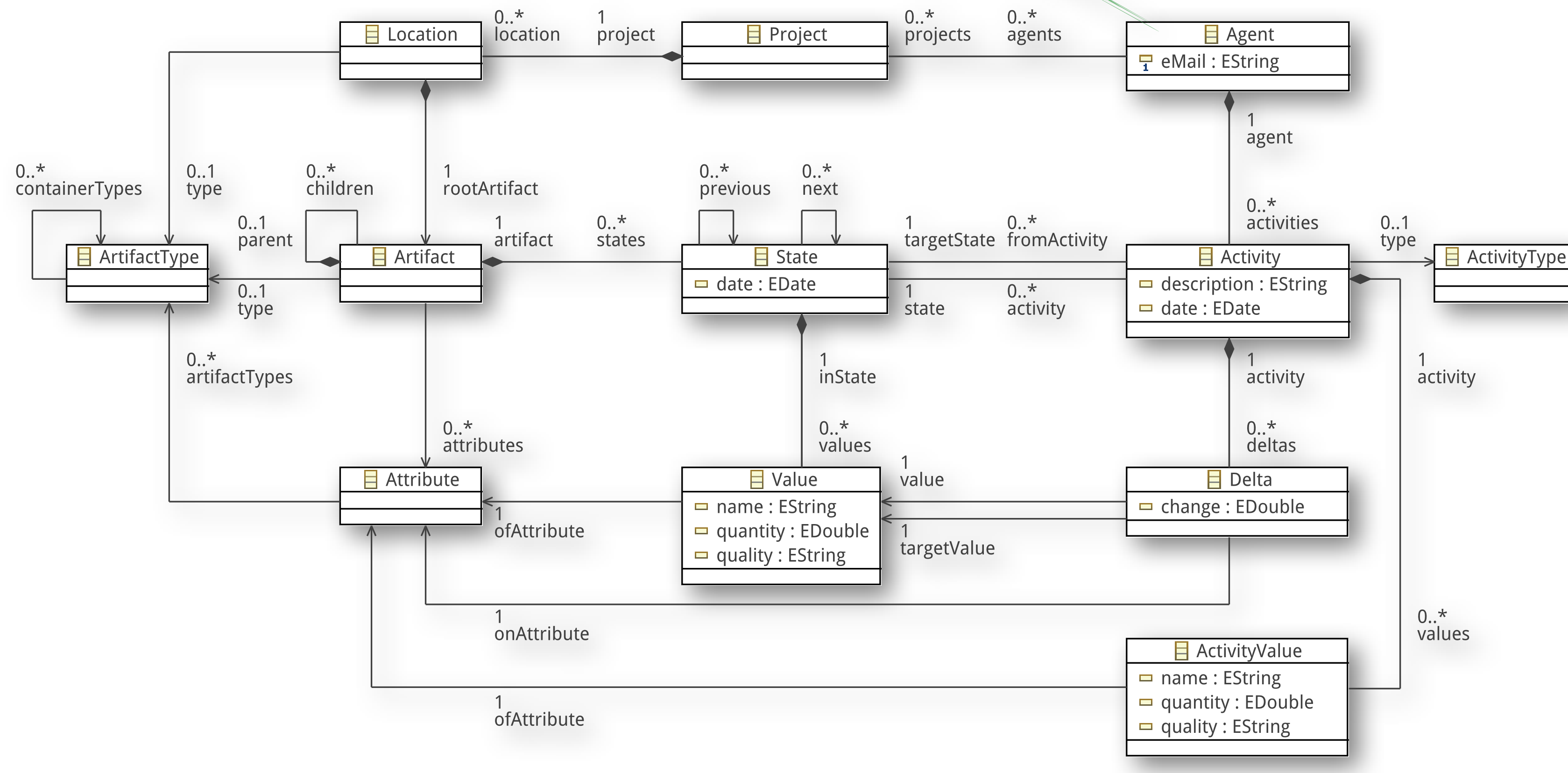

DEVELOPER-CENTRIC
Context
Change
9
DECENT INFRASTRUCTURE
Assessment Results
Assessment Application
Assessment Assets
Assessment Transformation
Assessment Model Instances
Facts Transformation
Facts Model Instances
Facts Translation
Facts Assets
Facts Extraction
Raw Assets
DECENT PREDICTION
39


Req
Law
Bug
Bug?
Change
Context

DECENT META-MODEL



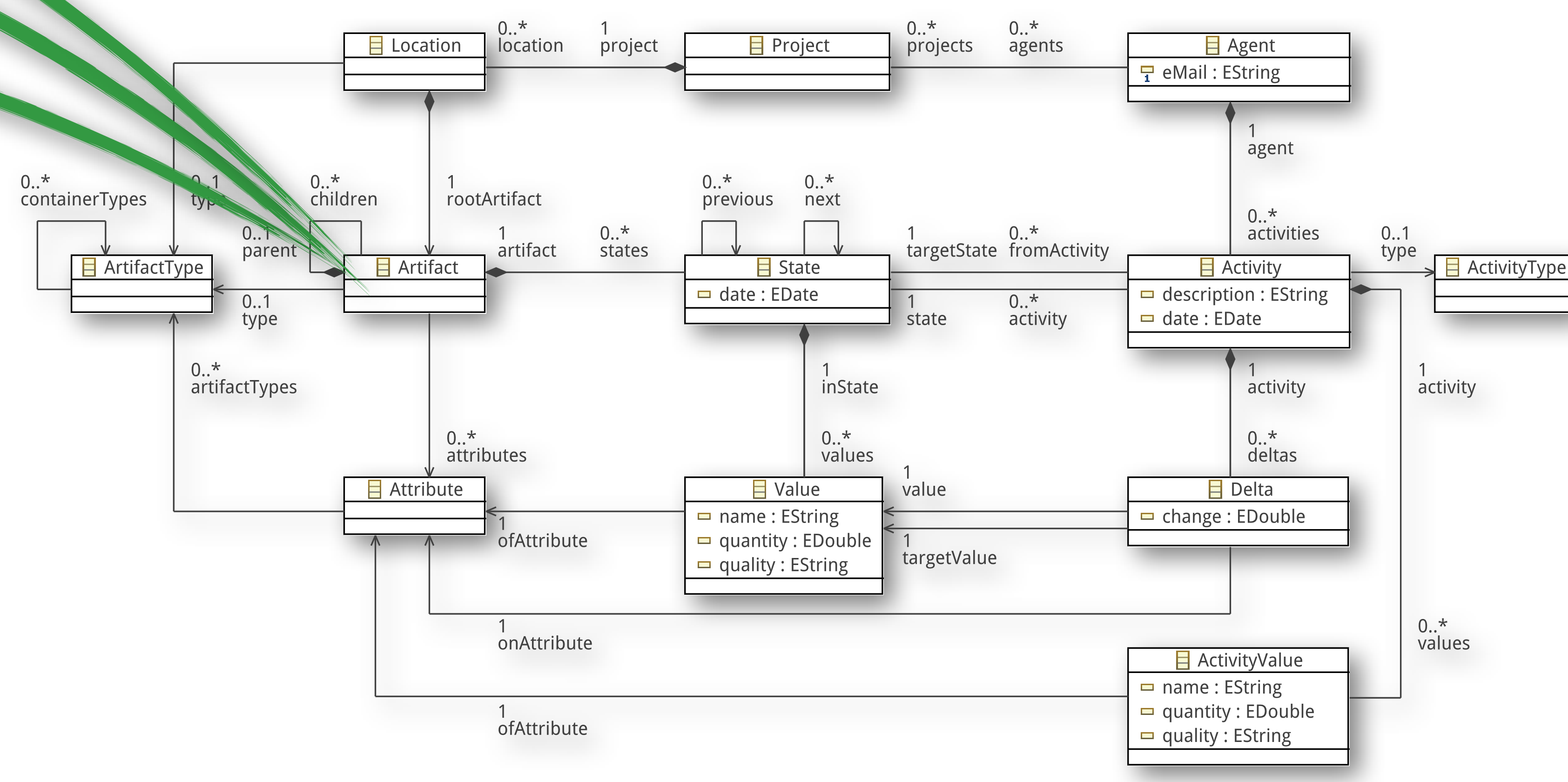
Model	Summary	Context(cc)	Delta(cc)	Context(loc)	Delta(loc)	Context(com)	Delta(com)	Hunks(	Addec	Removec	Own changes (past)	Tota
> lueck (10)	Commits: 1											
> matgic78 (22)	Commits: 59											
⌄ megabigbug (15)	Commits: 95											
⌄ 575	Files: 1											
⌄ settings.cpp		1	0	104	3	38	0	2	5	0	0	41
0:0=>53:53	Difference: 1											
0:0=>112:115	Difference: 4											
⌄ Affected Functions												
> Private(SettingsDialog *)	Hits:1	0	0	34	3	0	0	1				
⌄ 578	Files: 1											
⌄ webview.cpp		0	0	3	0	24	0	32	59	54	0	93
125:125=>125:125	Difference: 0											
0:0=>127:127	Difference: 1											
0:0=>130:131	Difference: 2											
133:134=>136:145	Difference: 8											
136:138=>0:0	Difference: -3											
141:145=>149:151	Difference: -2											
147:148=>0:0	Difference: -2											
150:151=>154:157	Difference: 2											
153:153=>159:160	Difference: 1											
0:0=>162:163	Difference: 2											
159:160=>0:0	Difference: -2											
162:163=>0:0	Difference: -2											
168:169=>0:0	Difference: -2											
171:172=>0:0	Difference: -2											
177:178=>0:0	Difference: -2											




```
    different context menus:
    -link actions
    -actions for a text selected in field
    -actions for a field
    -actions for text selected in a page
    -actions for a page

--- a/src/webview.cpp
+++ b/src/webview.cpp
```

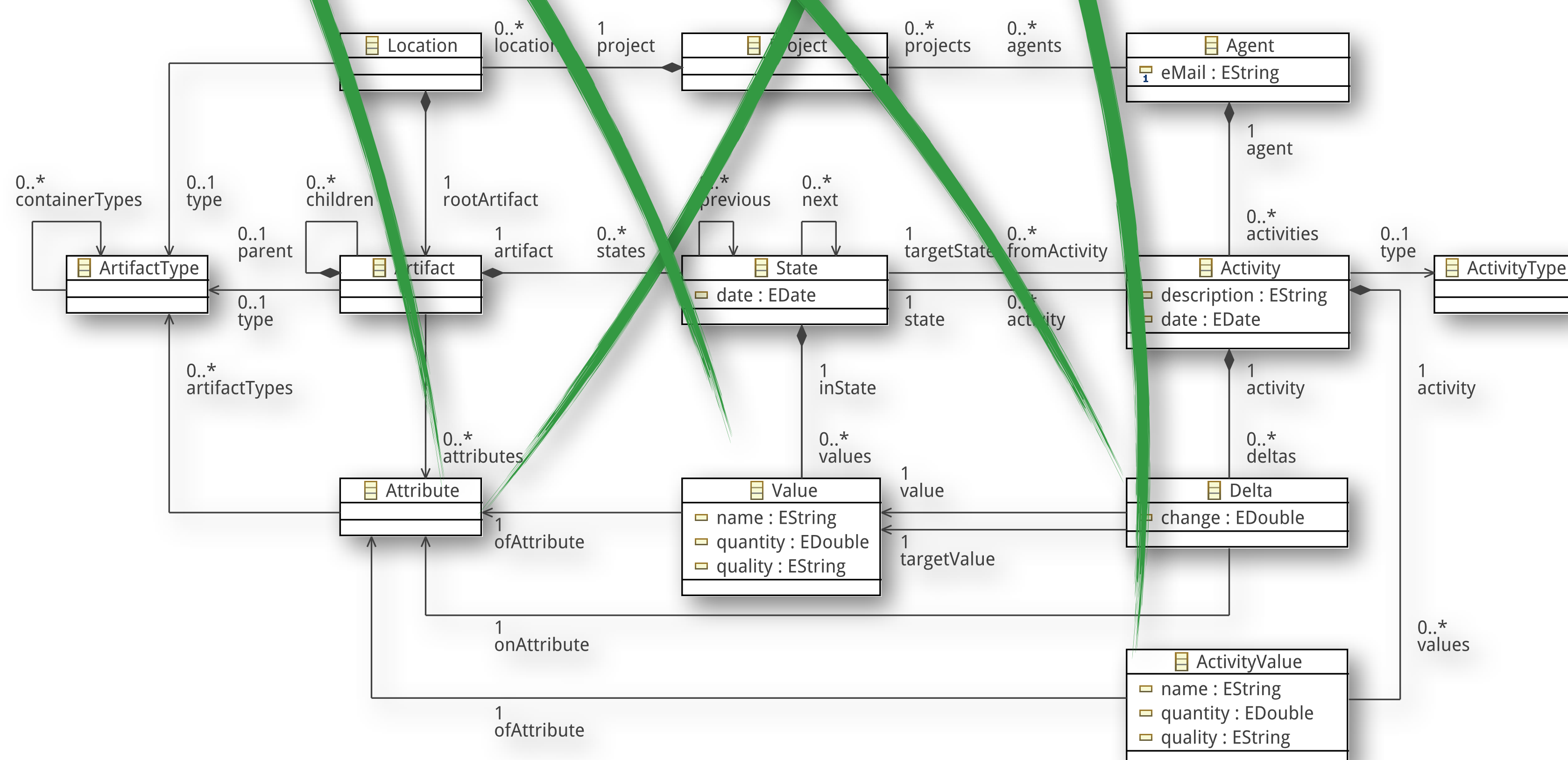
Model	Summary	Context(cc)	Delta(cc)	Context(loc)	Delta(loc)	Context(com)	Delta(com)	Hunks	Added	Removed	Own changes (past)	Total
> lueck (10)	Commits: 1											
> matgic78 (22)	Commits: 59											
⌄ megabigbug (15)	Commits: 95											
⌄ 575	Files: 1											
⌄ settings.cpp		1	0	104	3	38	0	2	5	0	0	41
0:0=>53:53	Difference: 1											
0:0=>112:115	Difference: 4											
⌄ Affected Functions												
> Private(SettingsDialog *)	Hunks: 1	0	0	34	3	0	0	1				
⌄ 578	Files: 1											
⌄ webview.cpp		[illegible]	0	3	0	24	0	32	59	54	0	93
125:125=>125:125	Difference: 0											
0:0=>127:127	Difference: 1											
0:0=>130:131	Difference: 2											
133:134=>136:145	Difference: 8											
136:138=>0:0	Difference: -3											
141:145=>149:151	Difference: -2											
147:148=>0:0	Difference: -2											
150:151=>154:157	Difference: 2											
153:153=>159:160	Difference: 1											
0:0=>162:163	Difference: 2											
159:160=>0:0	Difference: -2											
162:163=>0:0	Difference: -2											
168:169=>0:0	Difference: -2											
171:172=>0:0	Difference: -2											
177:178=>0:0	Difference: -2											




```
different context menus:
-link actions
-actions for a text selected in field
-actions for a field
-actions for text selected in a page
-actions for a page

--- a/src/webview.cpp
+++ b/src/webview.cpp
```

Model	Summary	Context(cc)	Delta(cc)	Context(loc)	Delta(loc)	Context(com)	Delta(com)	Hunks	Added	Removed	Own changes (past)	Total
> lueck (10)	Commits: 1											
> matgic78 (22)	Commits: 59											
⌄ megabigbug (15)	Commits: 95											
⌄ 575	Files: 1											
⌄ settings.cpp		1	0	104	3	38	0	2	5	0	0	41
0:0=>53:53	Difference: 1											
0:0=>112:115	Difference: 4											
⌄ Affected Functions												
> Private(SettingsDialog *)	Hits:1	0	0	34	3	0	0	1				
⌄ 578	Files: 1											
⌄ webview.cpp		0	0	3	0	24	[illegible]	32	[illegible]	54	0	93
125:125=>125:125	Difference: 0											
0:0=>127:127	Difference: 1											
0:0=>130:131	Difference: 2											
133:134=>136:145	Difference: 8											
136:138=>0:0	Difference: -3											
141:145=>149:151	Difference: -2											
147:148=>0:0	Difference: -2											
150:151=>154:157	Difference: 2											
153:153=>159:160	Difference: 1											
0:0=>162:163	Difference: 2											
159:160=>0:0	Difference: -2											
162:163=>0:0	Difference: -2											
168:169=>0:0	Difference: -2											
171:172=>0:0	Difference: -2											
177:178=>0:0	Difference: -2											




```
    different context menus:
    -link actions
    -actions for a text selected in field
    -actions for a field
    -actions for text selected in a page
    -actions for a page

--- a/src/webview.cpp
+++ b/src/webview.cpp
```

Model	Summary	Context(cc)	Delta(cc)	Context(loc)	Delta(loc)	Context(com)	Delta(com)	Hunks(	Addec	Removec	Own changes (past)	Tota
› 1702	Files: 1											
⌄ 1720	Files: 4											
⌄ previewselectorbar.cpp		4	0	75	-1	32	-1	1	0	3	0	11
127:129=>0:0	Difference: -3											
› Affected Functions												
⌄ tabbar.cpp		37	0	237	0	41	0	1	1	1	1	85
184:184=>184:184	Difference: 0											
› Affected Functions												
⌄ websnap.cpp		6	3	80	4	41	4	7	35	20	3	37
0:0=>82:105	Difference: 24											
84:84=>108:108	Difference: 0											
99:101=>123:124	Difference: -1											
103:106=>126:131	Difference: 2											
112:120=>137:137	Difference: -8											
123:124=>0:0	Difference: -2											
151:151=>166:166	Difference: 0											
⌄ Affected Functions												
› render(const QWebPage	Hits:1	1	0	9	0	2	0	1				
› renderTabPreview(const	Hits:1	1	0	5	0	0	0	1				
› saveResult(bool)	Hits:1	1	0	13	0	0	0	1				
› renderPagePreview(const	Hits:4	4	0	19	0	9	0	4				
› websnap.h		0	0	0	0	0	0	2	3	1	3	29
› 1721	Files: 1											
⌄ 1724	Files: 4											
› webpage.cpp		96	0	460	0	96	9	2	10	0	5	155
› websnap.cpp		9	3	84	18	45	6	9	43	10	4	38
› websnap.h		0	0	0	0	0	0	1	22	0	5	31

```
    different context menus:
    -link actions
    -actions for a text selected in field
    -actions for a field
    -actions for text selected in a page
    -actions for a page

--- a/src/webview.cpp
+++ b/src/webview.cpp
```

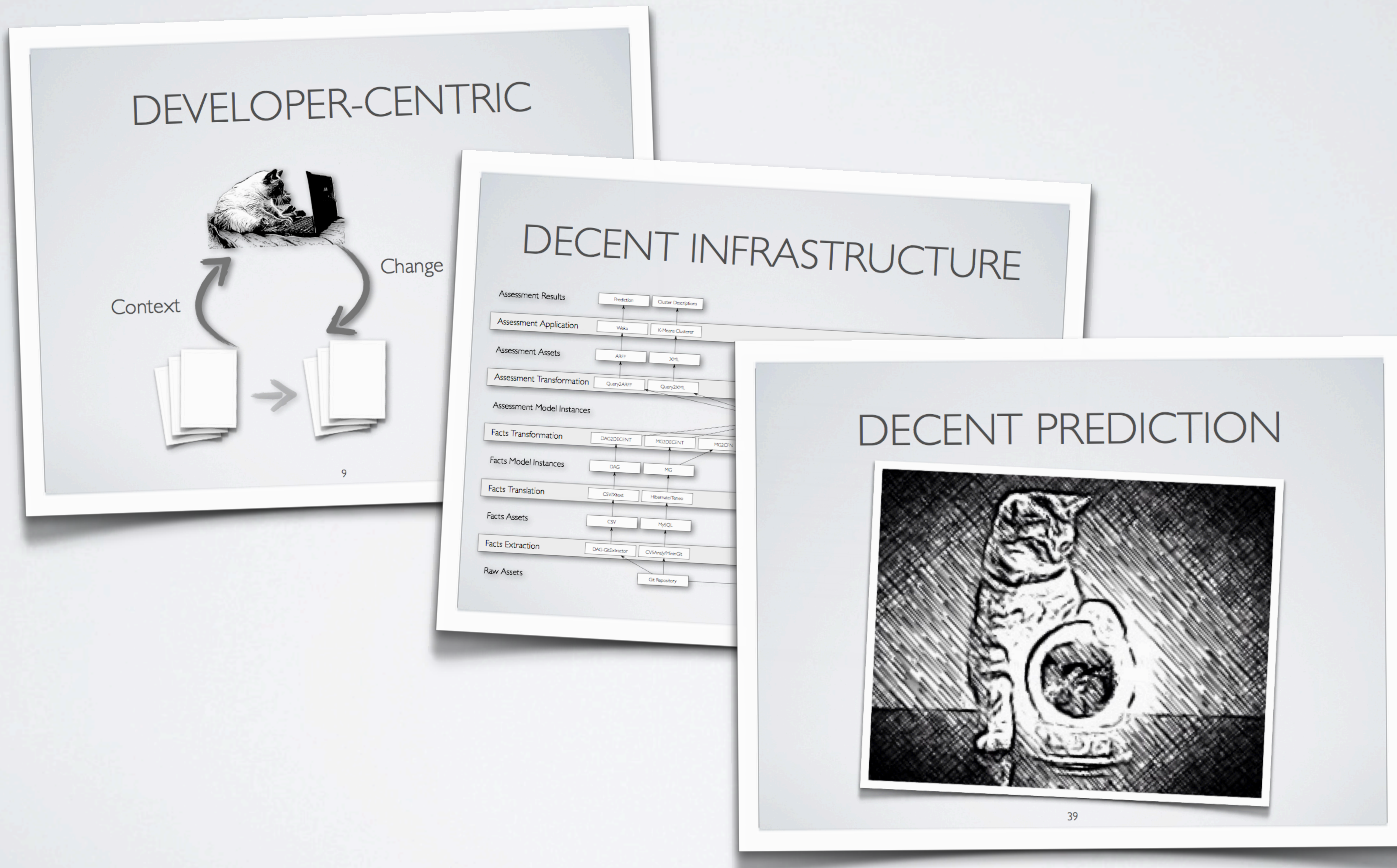
Total changes (past)	Ra	Ownership	Own fragments	Total fragments	Own fix commits	Total fix commits	Own bug commits	Total bug commits	Bug Confidence Weight	Fix	Bug	Clo
					0	1	0	0	1.0	0	1	
					1	2	1	9	0.5	1	1	
11					0	0	0	2	0.0	1	0	
85					0	1	0	1	0.16666666666666666	1	1	
37					1	1	1	8	1.0	1	1	
29					0	0	0	1	0.0	1	0	
					0	0	0	0	0.0	0	0	
					0	1	0	0	1.0	0	1	
155					0	1	0	0	0.5	0	1	
38					0	0	0	0	0.0	0	0	
31					0	0	0	0	0.0	0	0	

e Weight	Fix	Bug	Clone	Last change own	Distance to last own change	Distance to last	Average distance between ow	Average distance	Churn (added/loc)	Density (hunks/loc)
	0	1								
	1	1								
	1	0		false	-1	53	-1	0	0.0	0.013333333333333334
6666666	1	1		false	325	0	325	3	0.004219409282700	0.004219409282700422
	1	1		false	240	4	103	8	0.4375	0.0875
	1	0		false	240	4	103	10	0.0	0.0
	0	0								
	0	1								
	0	1		false	124	8	107	3	0.021739130434782	0.004347826086956522
	0	0		true	4	4	138	14	0.511904761904761	0.10714285714285714
	0	0		true	4	4	110	17	0.0	0.0

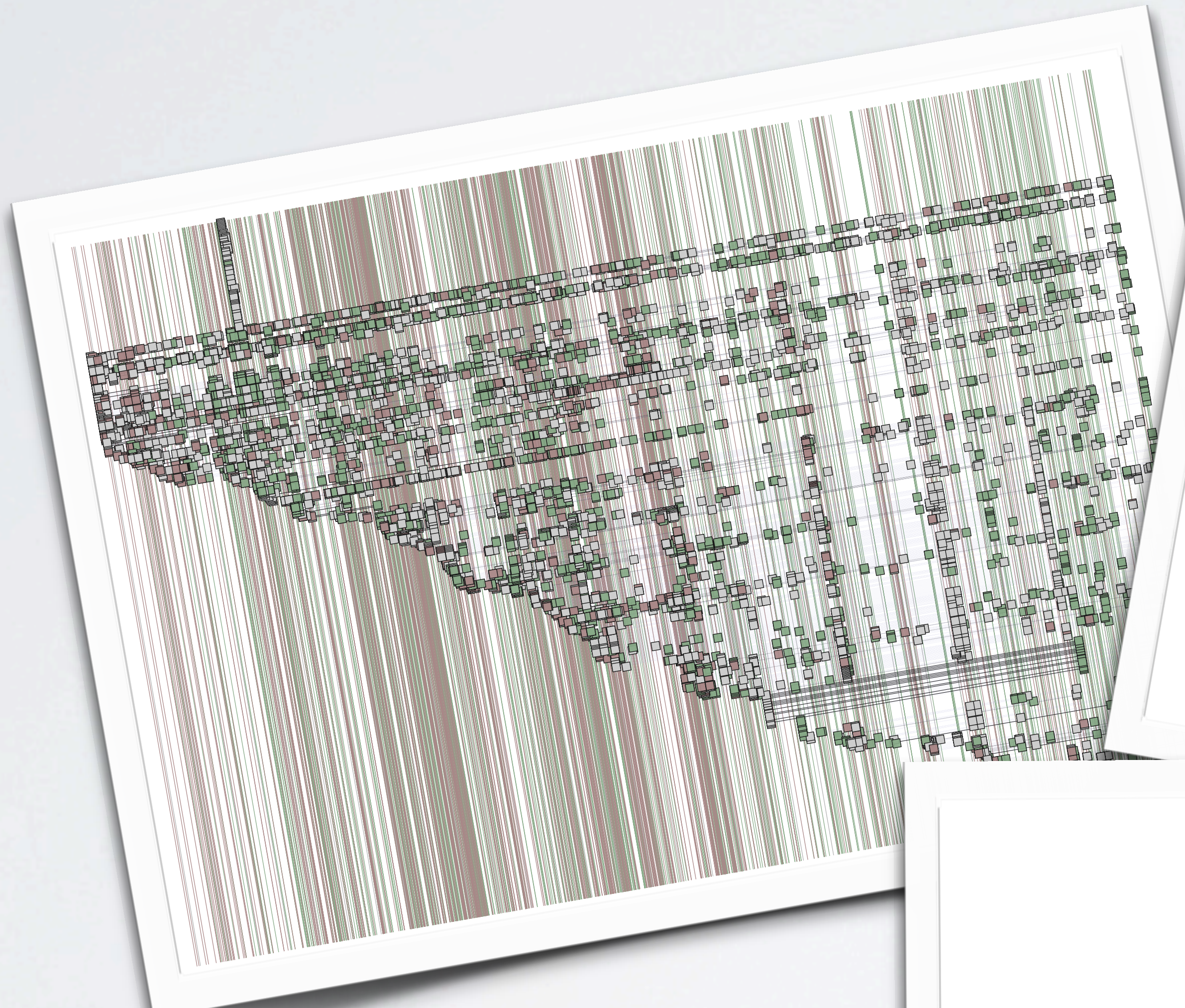
DECENT PREDICTION

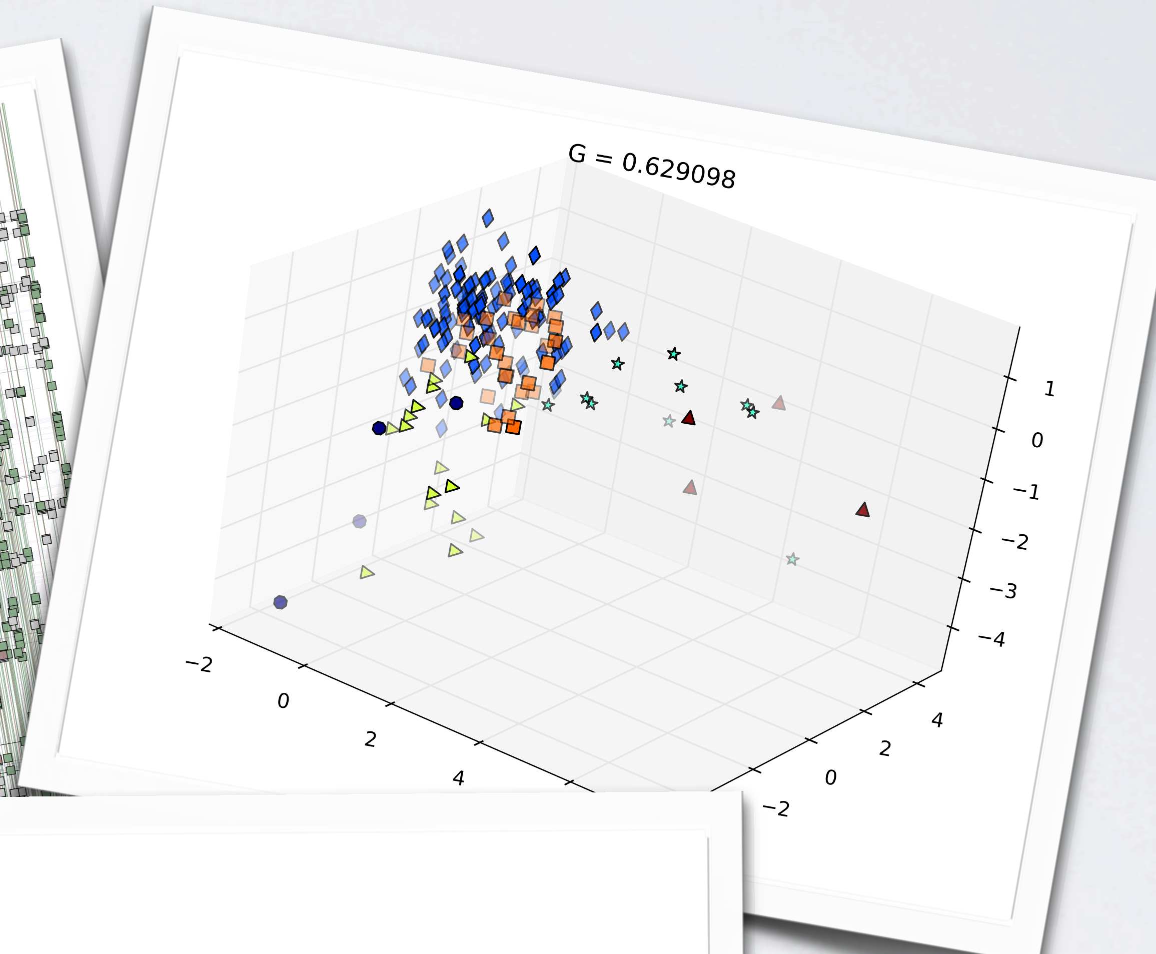
- Developers as first class citizens in software assessment
- Developer-specific factors contributing to risk
- Personalized and contextualized feedback
- Improvement of software assessment and quality

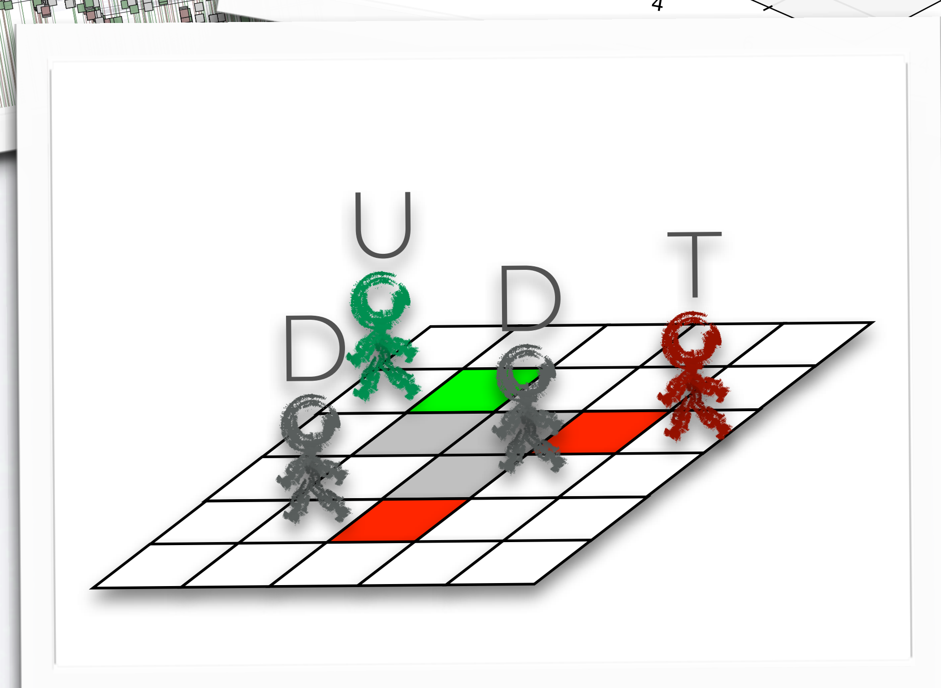
SUMMARY


DEVELOPER-CENTRIC
Context
Change
9
DECENT INFRASTRUCTURE
Assessment Results
Assessment Application
Assessment Assets
Assessment Transformation
Assessment Model Instances
Facts Transformation
Facts Model Instances
Facts Translation
Facts Assets
Facts Extraction
Raw Assets
DECENT PREDICTION
39

FURTHER APPLICATIONS








Microsoft

DEVELOPER-CENTRIC SOFTWARE ASSESSMENT

Philip Makedonski
makedonski@informatik.uni-goettingen.de

Jens Grabowski
grabowski@informatik.uni-goettingen.de

Software Engineering for Distributed Systems
Goldschmidtstr. 7
37077 Göttingen
Germany
www.swe.informatik.uni-goettingen.de

LITERATURE

- Girba, T., A. Kuhn, M. Seeberger, and S. Ducasse. 2005. "How Developers Drive Software Evolution." In Eighth International Workshop on Principles of Software Evolution, 113 – 122. doi:10.1109/IWPSE.2005.21.
- Hindle, A. 2011. "Evidence-based Software Process Recovery: A Post-doctoral View." In 2011 27th IEEE International Conference on Software Maintenance (ICSM), 562–567. doi:10.1109/ICSM.2011.6080831.
- Hindle, Abram. 2010. "Evidence-based Software Process Recovery". PhD Thesis, University of Waterloo.
- Nierstrasz, Oscar. 2012. "Agile Software Assessment with Moose." SIGSOFT Softw. Eng. Notes 37 (3) (May): 1–5. doi: 10.1145/2180921.2180925.
- Shihab, Emad, Ahmed E. Hassan, Bram Adams, and Zhen Ming Jiang. 2012. "An Industrial Study on the Risk of Software Changes." In Proceedings of the ACM SIGSOFT 20th International Symposium on the Foundations of Software Engineering, 62:1–62:11. FSE '12. New York, NY, USA: ACM. doi:10.1145/2393596.2393670. http://doi.acm.org/10.1145/2393596.2393670.
- Weyuker, Elaine J., Thomas J. Ostrand, and Robert M. Bell. 2007. "Using Developer Information as a Factor for Fault Prediction." In Proceedings of the Third International Workshop on Predictor Models in Software Engineering, 8. IEEE Computer Society.
- Makedonski, Philip, Fabian Sudau, and Jens Grabowski. 2013. "Towards a Model-Based Software Mining Infrastructure." To Appear in Proceedings of the 2nd International Workshop on Software Mining, Silicon Valley, CA, USA.

FURTHER LITERATURE

- H. C. Benestad, B. Anda, and E. Arisholm, "A systematic review of empirical software engineering studies that analyze individual changes — simula.no," 2008.
- A. Schröter, T. Zimmermann, R. Premraj, and A. Zeller, "If your bug database could talk," in In Proceedings of the 5th International Symposium on Empirical Software Engineering, Volume II: Short Papers and Posters, 2006, pp. 18–20.
- A. Schröter, T. Zimmermann, R. Premraj, and A. Zeller, "Where do bugs come from?," SIGSOFT Softw. Eng. Notes, vol. 31, no. 6, pp. 1–2, Nov. 2006.
- T. Girba, A. Kuhn, M. Seeberger, and S. Ducasse, "How developers drive software evolution," in Eighth International Workshop on Principles of Software Evolution, 2005, pp. 113 – 122.
- S. Wagner, K. Lochmann, L. Heinemann, M. Kläs, A. Trendowicz, R. Plösch, A. Seidl, A. Goeb, and J. Streit, "The quamoco product quality modelling and assessment approach," in Proceedings of the 2012 International Conference on Software Engineering, Piscataway, NJ, USA, 2012, pp. 1133–1142.

FURTHER LITERATURE

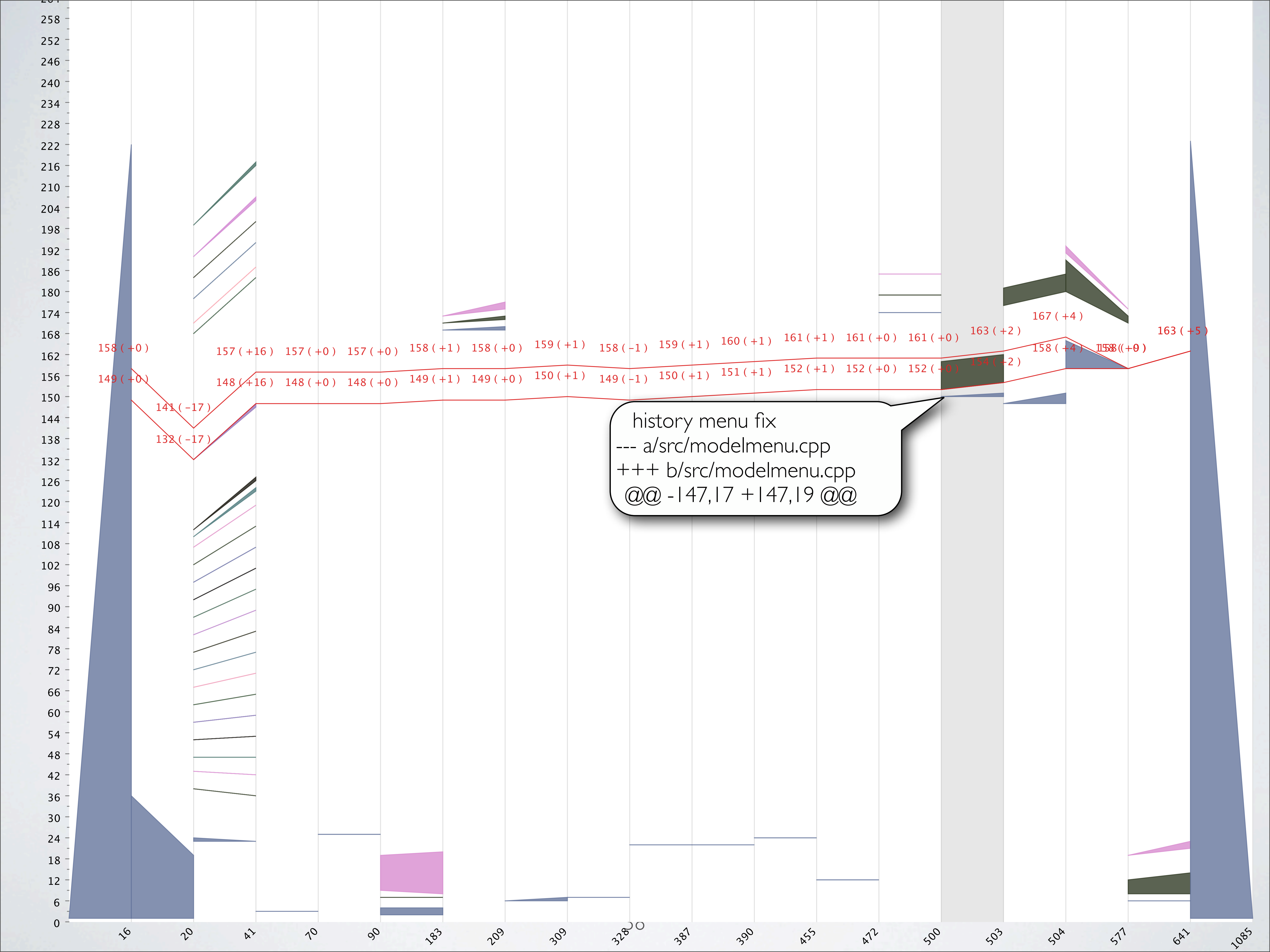
- J. Ekanayake, J. Tappolet, H. Gall, and A. Bernstein, "Time variance and defect prediction in software projects," Empirical Software Engineering, vol. 17, no. 4, pp. 348–389, 2012.
- R. Moser, W. Pedrycz, and G. Succi, "A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction," in Proceedings of the 30th international conference on Software engineering, New York, NY, USA, 2008, pp. 181–190.
- S. Olbrich, D. S. Cruzes, V. Basili, and N. Zazworka, "The evolution and impact of code smells: A case study of two open source systems," in 2009 3rd International Symposium on Empirical Software Engineering and Measurement, Lake Buena Vista, FL, USA, 2009, pp. 390–400.
- Sunghun Kim, T. Zimmermann, Kai Pan, and E. J. Whitehead, "Automatic Identification of Bug-Introducing Changes," in Automated Software Engineering, 2006. ASE '06. 21st IEEE/ACM International Conference on, 2006, pp. 81–90.
- A. Hindle, "Evidence-based Software Process Recovery," PhD Thesis, University of Waterloo, 2010.

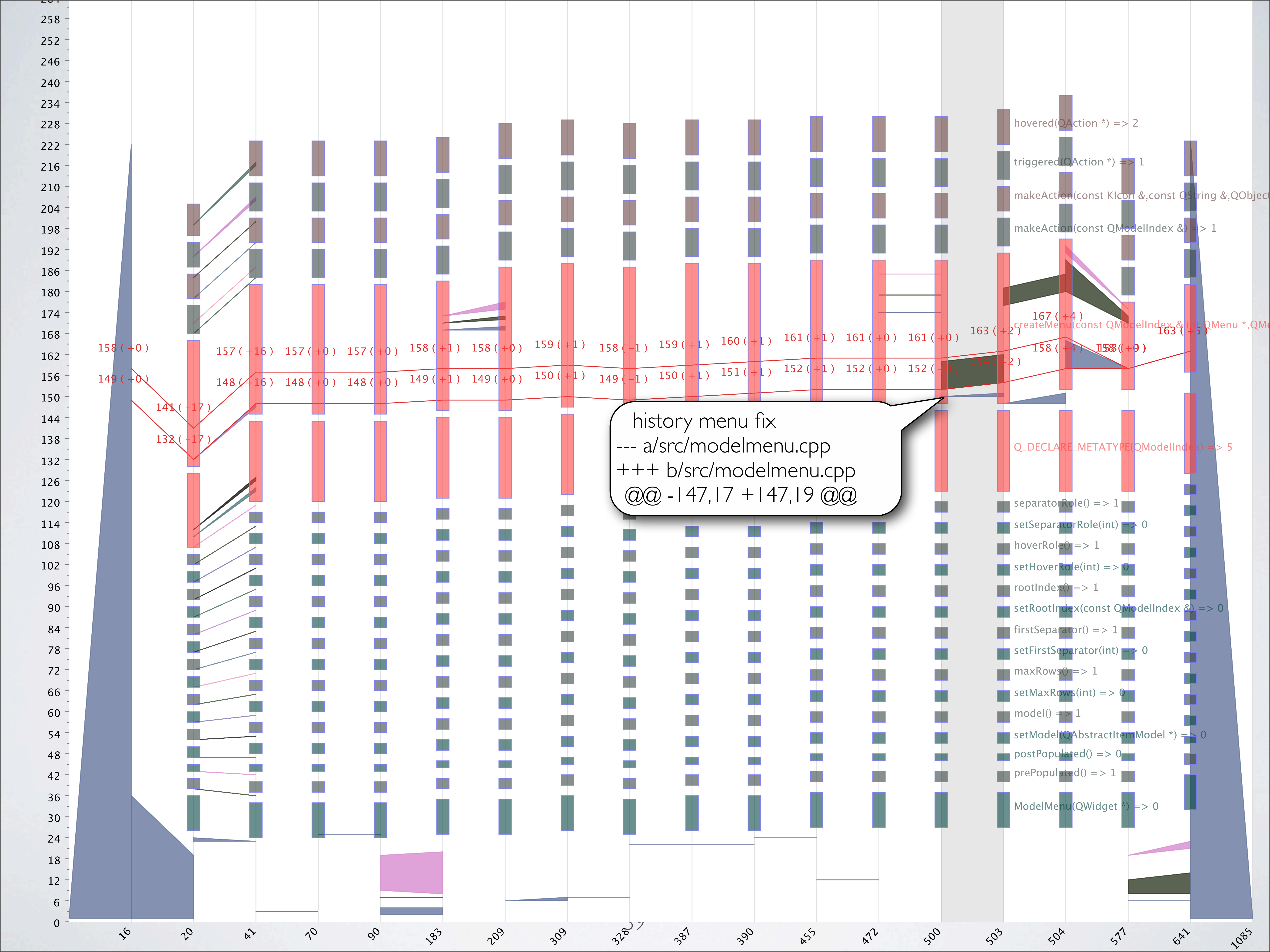
FURTHER LITERATURE

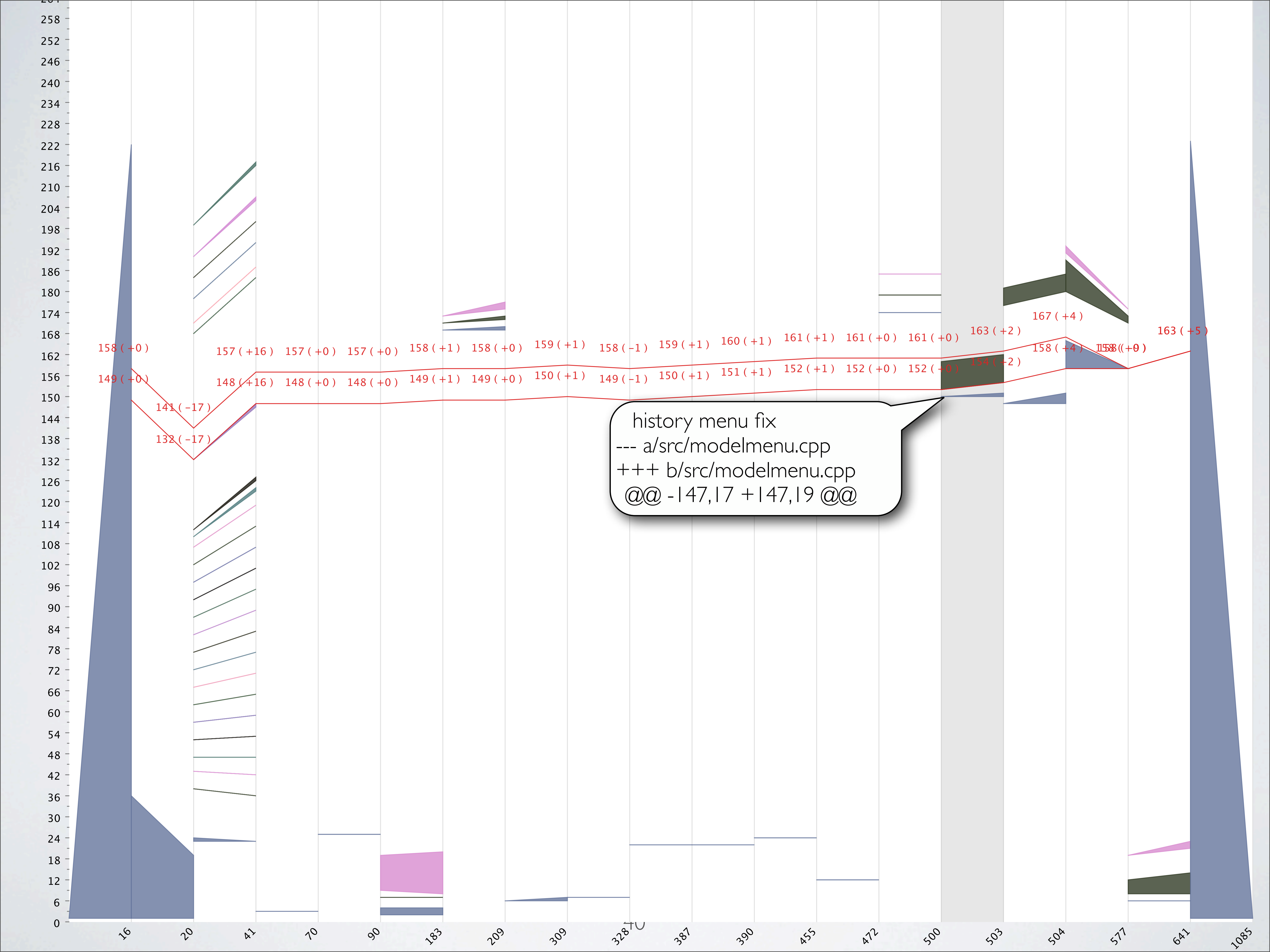
- E. J. Weyuker, T. J. Ostrand, and R. M. Bell, "Using Developer Information as a Factor for Fault Prediction," in Proceedings of the Third International Workshop on Predictor Models in Software Engineering, 2007, p. 8.
- C. Görg and P. Weißgerber, "Error detection by refactoring reconstruction," in Proceedings of the 2005 international workshop on Mining software repositories, St. Louis, Missouri, 2005, pp. 1–5.
- J. Śliwerski, T. Zimmermann, and A. Zeller, "When do changes induce fixes?," in Proceedings of the 2005 international workshop on Mining software repositories, St. Louis, Missouri, 2005, pp. 1–5.
- A. Tarvo, "Using Statistical Models to Predict Software Regressions," in Software Reliability Engineering, 2008. ISSRE 2008. 19th International Symposium on, 2008, pp. 259–264.
- A. Mockus and D. M. Weiss, "Predicting risk of software changes," Bell Labs Tech. J., vol. 5, no. 2, pp. 169–180, Apr. 2000.

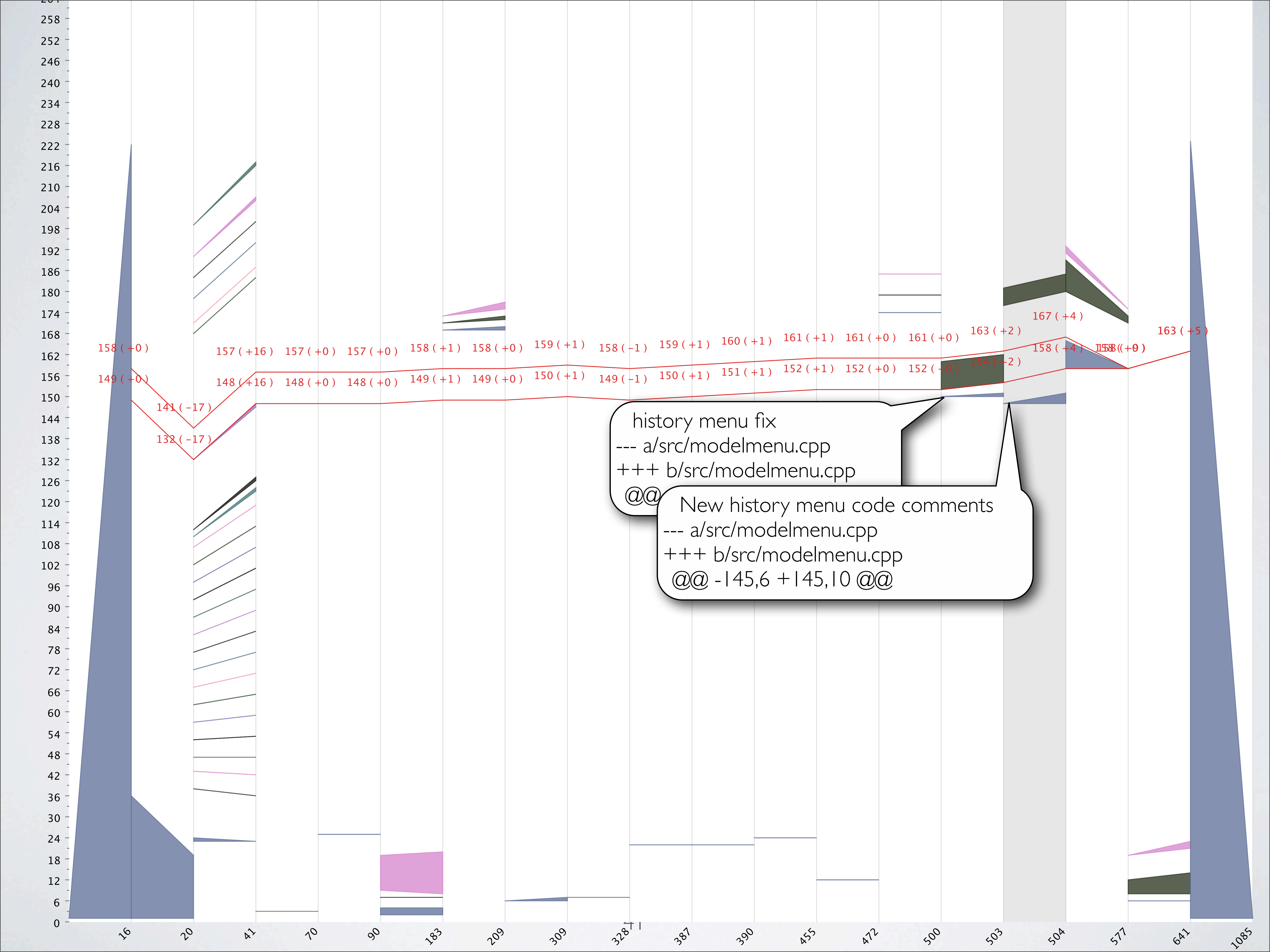
FURTHER LITERATURE

- M. Lungu, J. Malnati, and M. Lanza, "Visualizing Gnome with the Small Project Observatory," in Mining Software Repositories, 2009. MSR '09. 6th IEEE International Working Conference on, 2009, pp. 103–106.
- S. Kim, T. Zimmermann, E. J. Whitehead Jr., and A. Zeller, "Predicting Faults from Cached History," in Proceedings of the 29th international conference on Software Engineering, Washington, DC, USA, 2007, pp. 489–498.
- R. Premraj and K. Herzig, "Network Versus Code Metrics to Predict Defects: A Replication Study," in Empirical Software Engineering and Measurement (ESEM), 2011 International Symposium on, 2011, pp. 215 –224.
- T. Zimmermann and N. Nagappan, "Predicting defects using network analysis on dependency graphs," in Proceedings of the 30th international conference on Software engineering, New York, NY, USA, 2008, pp. 531–540.
- M. Pinzger, N. Nagappan, and B. Murphy, "Can developer-module networks predict failures?," in Proceedings of the 16th ACM SIGSOFT International Symposium on Foundations of software engineering, New York, NY, USA, 2008, pp. 2–12.

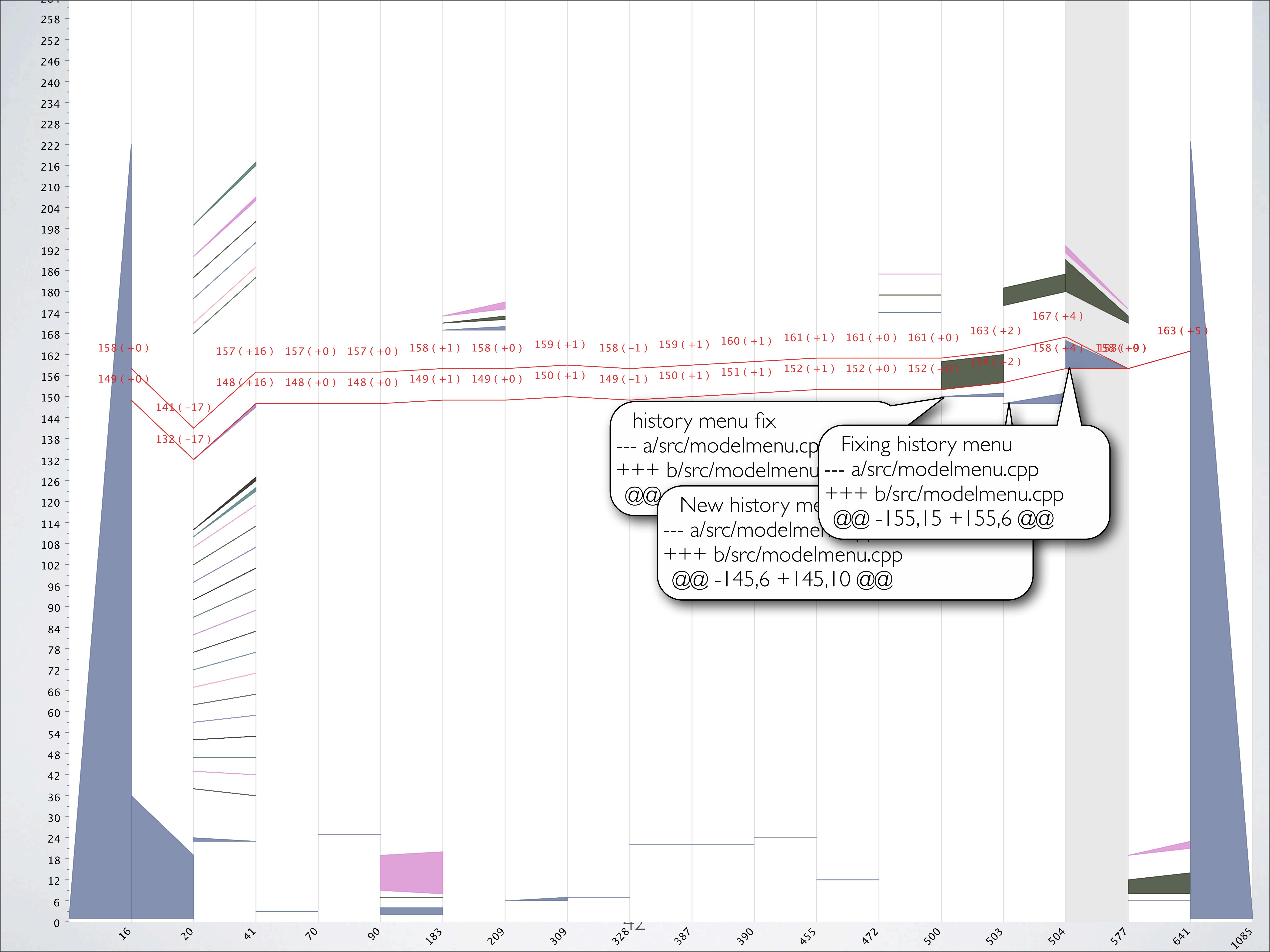

history menu fix
--- a/src/modelmenu.cpp
+++ b/src/modelmenu.cpp
@@ -147,17 +147,19 @@
16 20 41 70 90 183 209 309 328 387 390 455 472 500 503 504 577 641 1085


history menu fix
--- a/src/modelmenu.cpp
+++ b/src/modelmenu.cpp
@@ -147,17 +147,19 @@
hovered(QAction *) => 2
triggered(QAction *) => 1
makeAction(const KIcon &,const QString &,QObject
makeAction(const QModelIndex &) => 1
createMenu(const QModelIndex &,int,QMenu *,QMe
Q_DECLARE_METATYPE(QModelIndex) => 5
separatorRole() => 1
setSeparatorRole(int) => 0
hoverRole() => 1
setHoverRole(int) => 0
rootIndex() => 1
setRootIndex(const QModelIndex &) => 0
firstSeparator() => 1
setFirstSeparator(int) => 0
maxRows() => 1
setMaxRows(int) => 0
model() => 1
setModel(QAbstractItemModel *) => 0
postPopulated() => 0
prePopulated() => 1
ModelMenu(QWidget *) => 0
16
20
41
70
90
183
209
309
328
387
390
455
472
500
503
504
577
641
1085


history menu fix
--- a/src/modelmenu.cpp
+++ b/src/modelmenu.cpp
@@ -147,17 +147,19 @@
158 (+0)
149 (+0)
141 (-17)
132 (-17)
157 (+16)
148 (+16)
157 (+0)
148 (+0)
158 (+1)
149 (+1)
159 (+1)
150 (+1)
158 (-1)
149 (-1)
160 (+1)
151 (+1)
161 (+1)
152 (+1)
161 (+0)
152 (+0)
163 (+2)
154 (+2)
167 (+4)
158 (+4)
163 (+5)



history menu fix
--- a/src/modelmenu.cpp
+++ b/src/modelmenu.cpp
New history menu code comments
--- a/src/modelmenu.cpp
+++ b/src/modelmenu.cpp
@@ -145,6 +145,10 @@
16 20 41 70 90 183 209 309 328 387 390 455 472 500 503 504 577 641 1085



history menu fix
New history menu
+++ b/src/modelmenu.cpp
@@ -145,6 +145,10 @@
Fixing history menu
--- a/src/modelmenu.cpp
+++ b/src/modelmenu.cpp
@@ -155,15 +155,6 @@
16
20
41
70
90
183
209
309
328
387
390
455
472
500
503
504
577
641
1085