Tools for Software Engineers

Visual Studio Team Services

Microsoft

From Code Via Builds to Tests and Back
to Capture the Full Picture

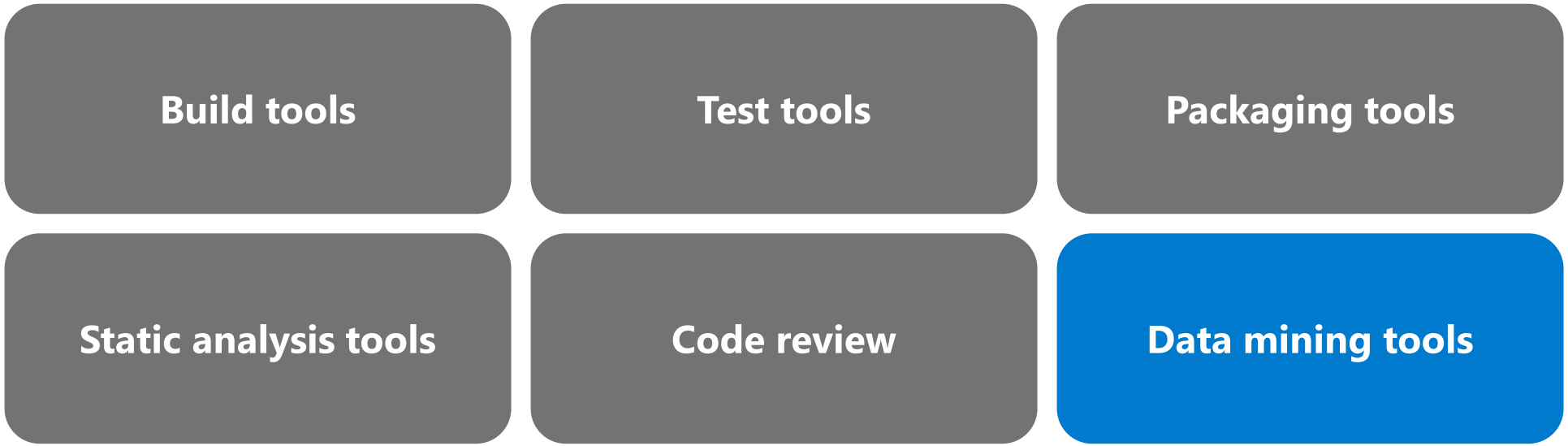
Kim Herzig (Software engineer & Researcher)

kimh@microsoft.com
www.research.microsoft.com/people/kimh
www.kim-herzig.de

Tools for Software Engineers

Visual Studio Team Services

Microsoft


Build tools
Test tools
Packaging tools
Static analysis tools
Code review
Data mining tools

When I joined Microsoft

Tools for Software Engineers

Visual Studio Team Services

Microsoft

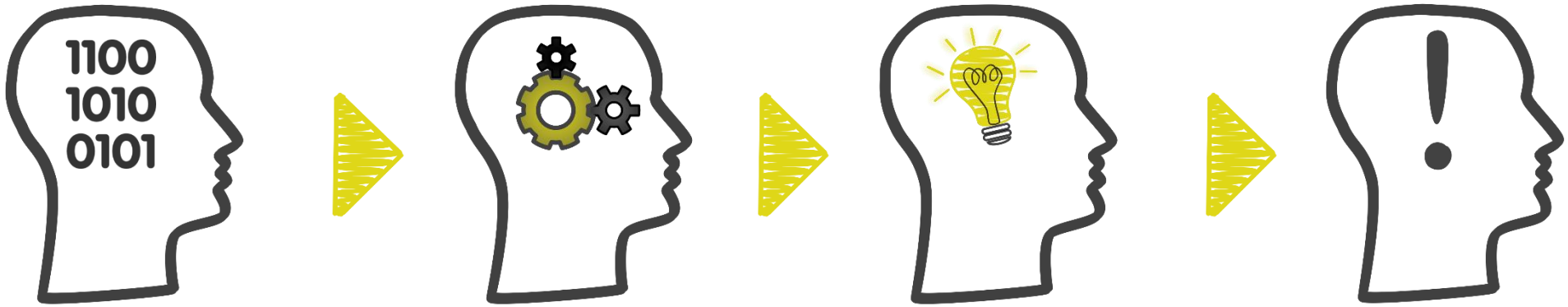
From Code Via Builds to Tests and back
to Capture the Full Picture

Kim Herzig (Software Engineer & Researcher)

kimh@microsoft.com
www.research.microsoft.com/people/kimh
www.kim-herzig.de

Mining Software Archives

"The Mining Software Repositories (MSR) field ***analyzes*** *the rich* ***data*** *available in software repositories* ***to uncover interesting and actionable information*** *about software systems and projects"* [MSR website]



Data → Thinking → Idea → Solution!

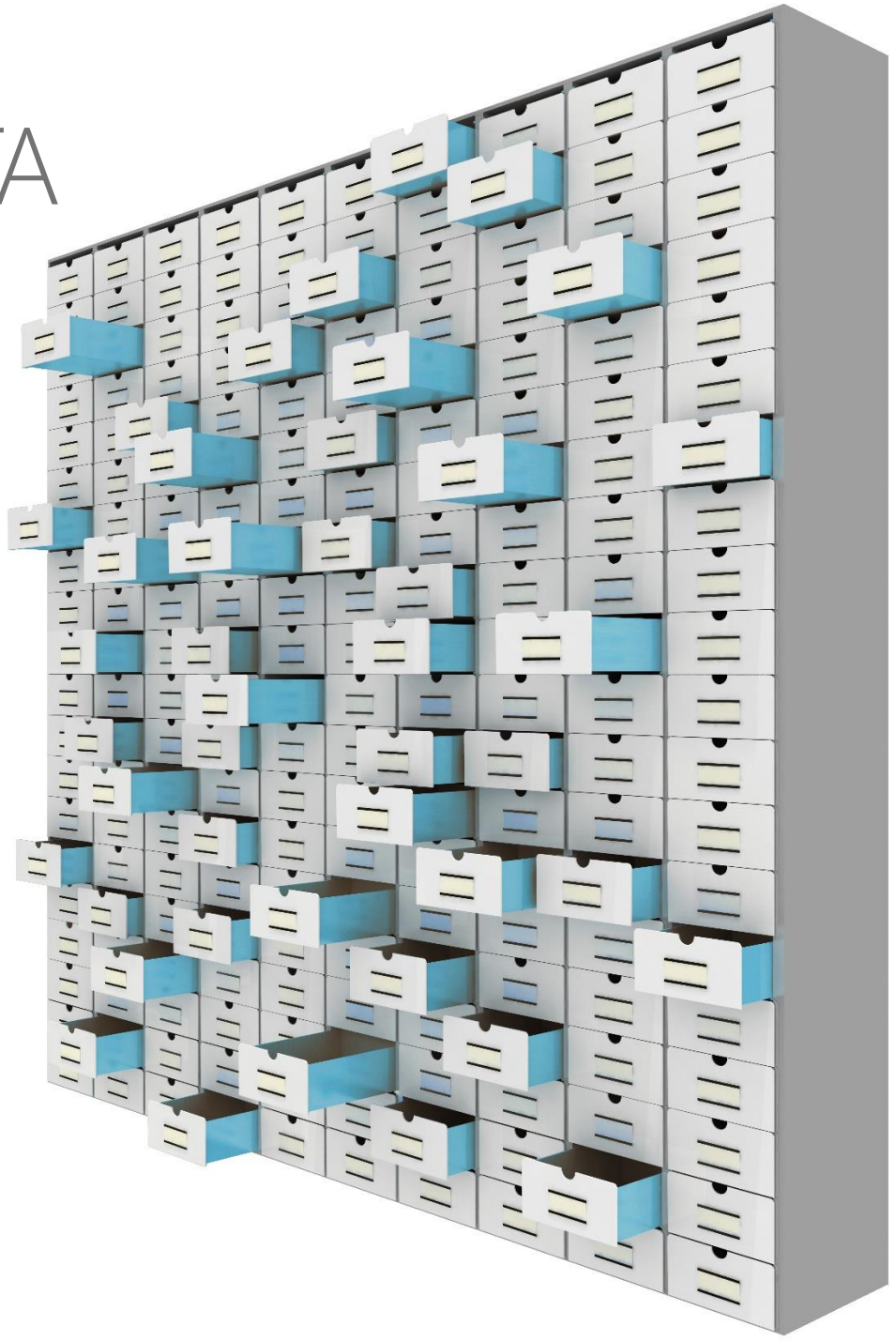
Do you know what you do?

- **What is the problem** you are trying to solve?
  - Hypothesis
  - Research Questions
- Is the **data appropriate**?
  - How should the data look like?
  - What do you expect?
- What's the **return of investment** of your solution?

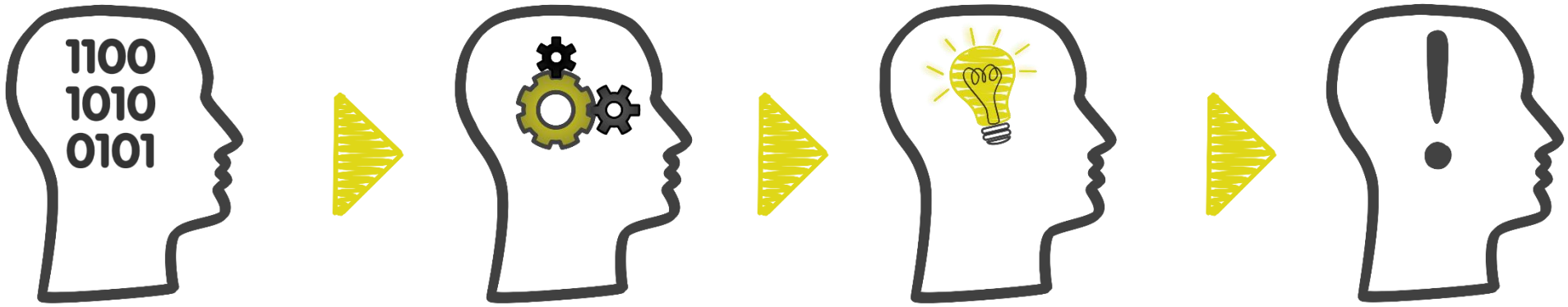
Don't Get Lost In Data

Mining @Microsoft

- Source code
- Bug report / crashes / stack traces
- Organizational structure
- Releases
- Test executions
- Code reviews
- Builds
- Desktop tracking / telemetry
- …



Mining Software Archives



Data → Thinking → Idea → Solution!

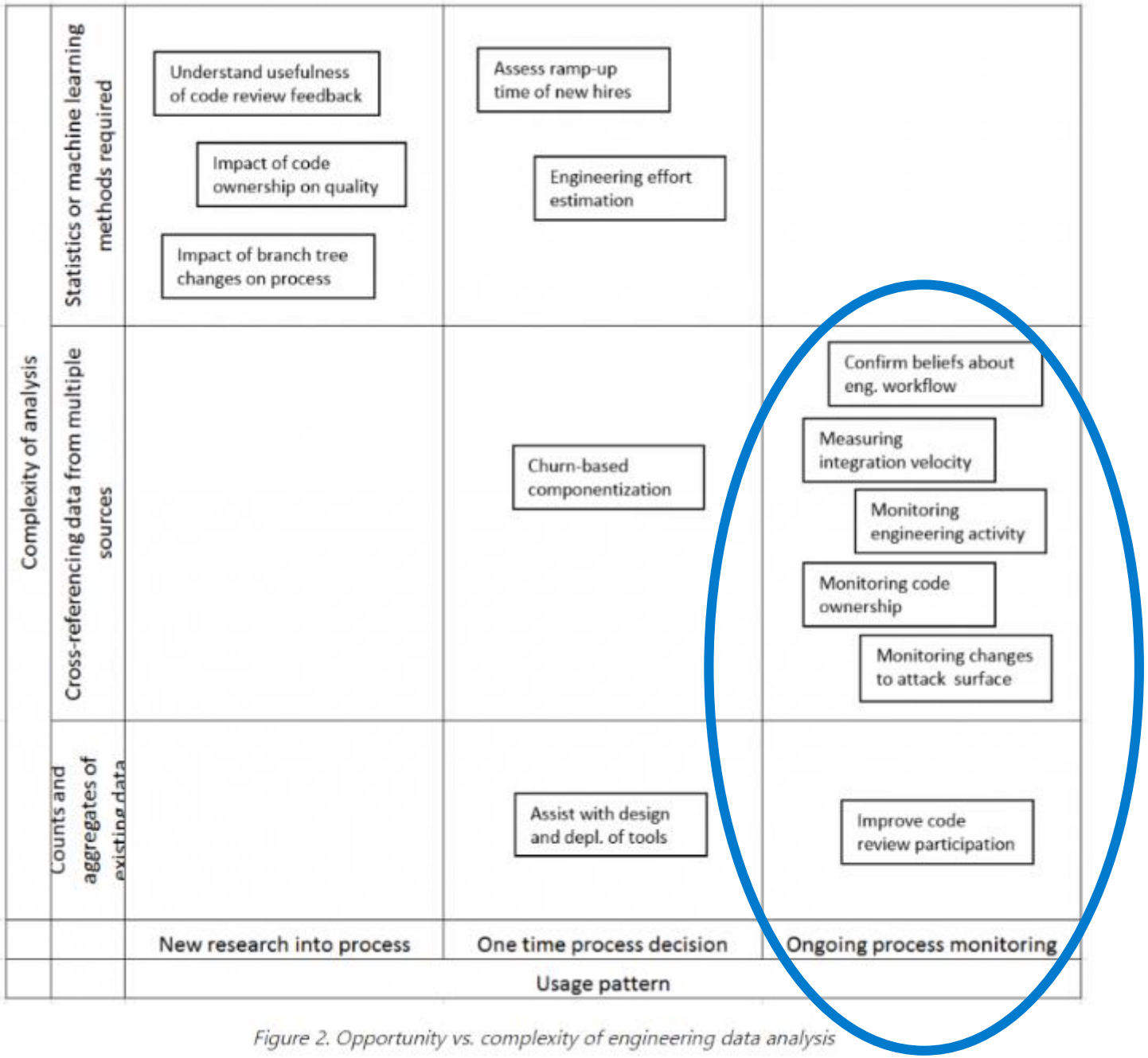
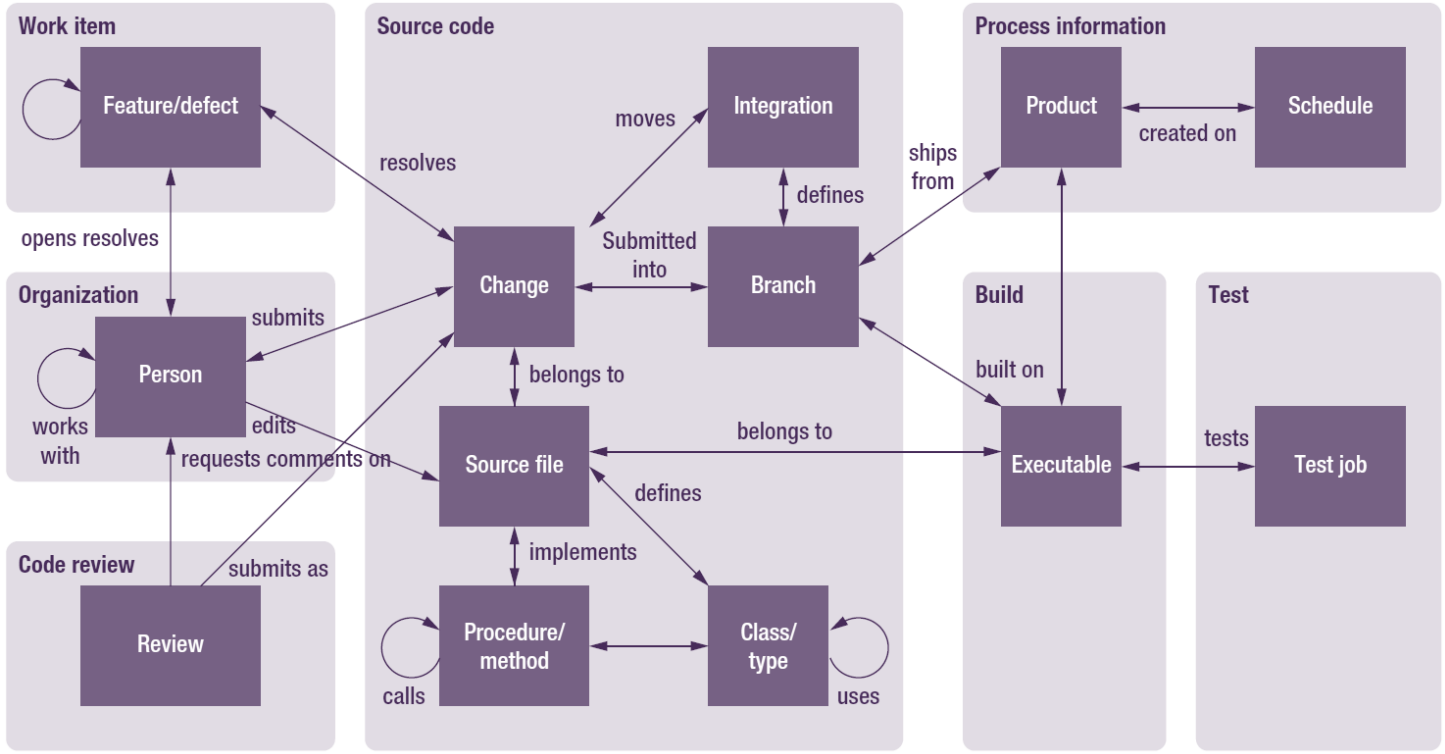


Figure 2. Opportunity vs. complexity of engineering data analysis

MSR @ MICROSOFT (CODEMINE / CLOUDMINE)

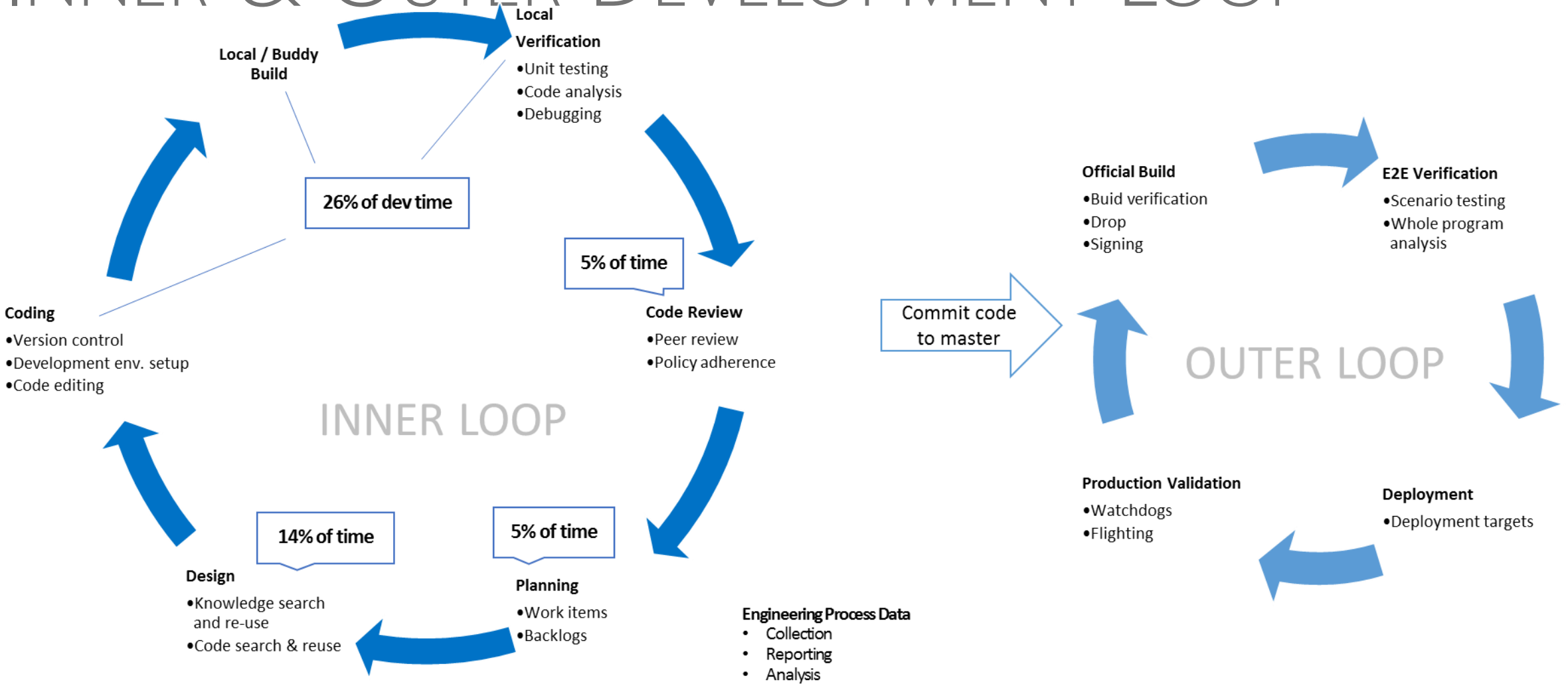
Measuring & benchmarking the entire development process: from change to deploy.

- Code velocity
- Code quality
- Code/test ownership
- Legacy code
- Branching structures
- …



*CodeMine, Czerwonka et. al., 2014, IEEE Software, 2013

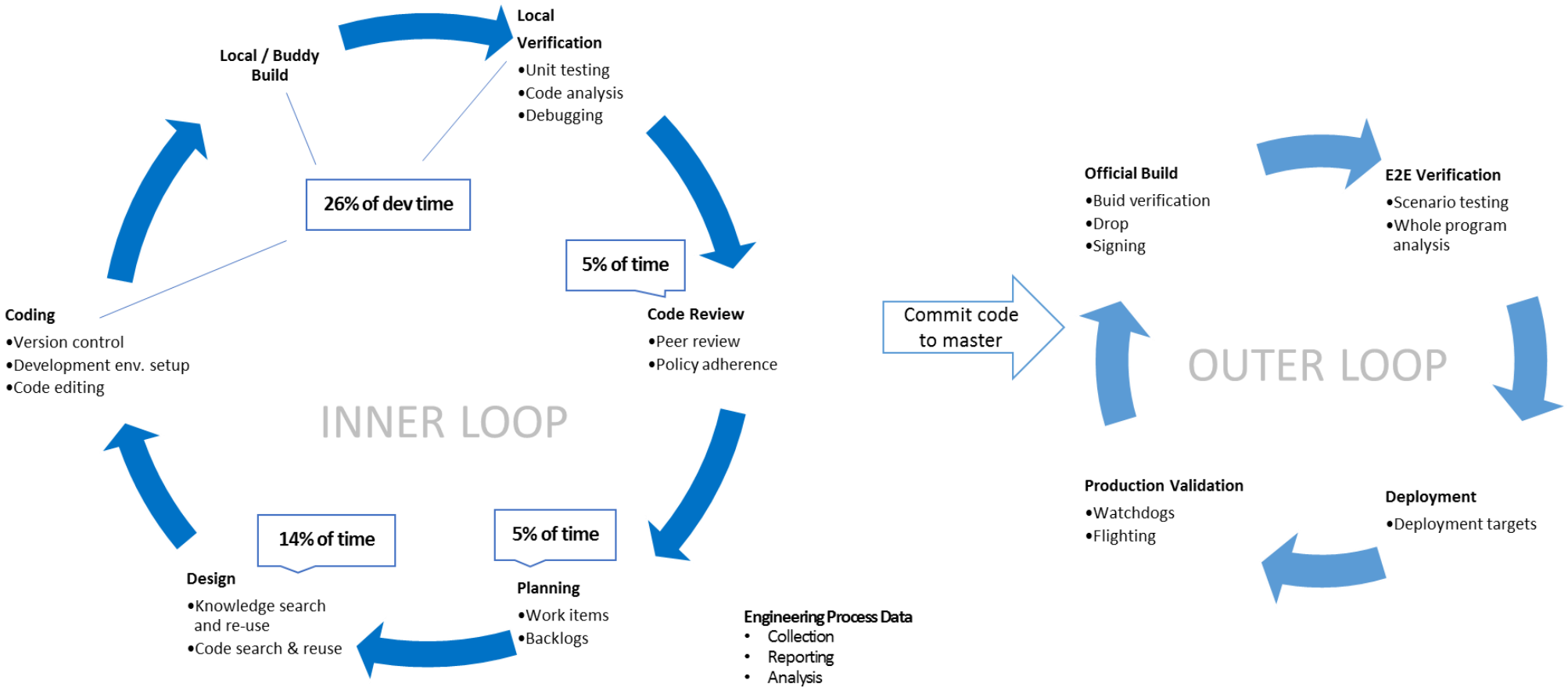
Inner & Outer Development Loop



Pre-check-in

Integration process

HIGH **EFFORT/COST**



MOST **RESEARCH**

What you can do with mining …

<video removed>

0

Pre-Phase

Data Scientist Play-Book

! Know your problem

- There is a difference between symptoms and problems

! Write down your assumptions

- We all make assumptions and it is important to make them explicit
- Share them with engineers.

! Engineers are humans

- They game the system
- Validate their statements with data and confront them with reality.

! Do not trust data

- Iterate over data with engineers
- Engineers claiming wrong data: 99% chance of wrong assumptions

! You will never get it right

- Accuracy of 80% is good. 100% is impossible.
- If you reach 90% do not trust you assumptions / data / results.

1

The source of all evil

Code Churn

Version Control <> Version Control

Centralized

+ Data on server
+ Full data control
+ No loss of information
- Dev needs to be on network
- All commits “public”
- Limited isolation

Distributed

+ Great work isolation
+ Fully developer controlled
+ More dev satisfaction
- Local dev data lost
- Historic data might be inaccurate
- Hard to mine entire dev process

The Basics

Thanks to Sascha Just (just@st.cs.uni-saarland.de)

Getting all changelists and their meta-data in the repository in efficient, machine readable format.

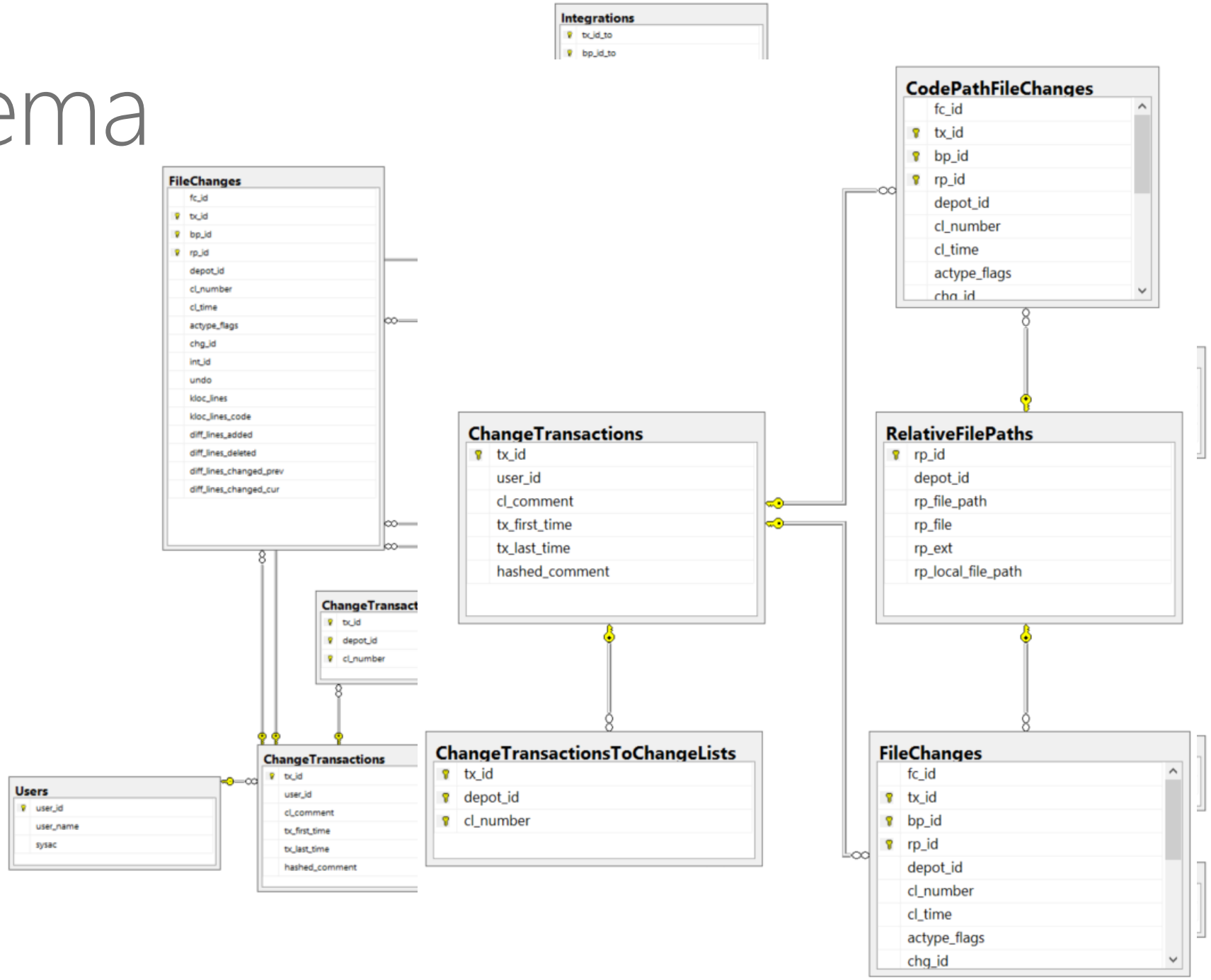
We are completely avoiding regular expressions. There is no need for it and in some languages (like Java) the implementation tends to be horrendously slow.

```
> git log --branches --remotes --tags --topo-order --find-copies --raw --numstat --no-abbrev --reverse --format=%n%H%n%T%n%an%n%ae%n%at%n%cn%n%ce%n%ct%n%s%n%b%n
```

```
1da198d551de42c41fe96810152274387daed102
914f79e0cc22a4fdcc11a84f4cac4781099bd0aa
Kim Herzig
kimh@microsoft.com
1454546288
Kim Herzig
kimh@microsoft.com
1454546288
Fixing argument conflict
```

```
:100644 100644 6c5a8f04126020b4726821601817230415b6d951 ad0b15c296ec10ae22aabd7c4034a1aef39febae M	src/CloudBuildMiner/Configuration/Configuration.cs
:100644 100644 bf53ccf35bf761d166d9a3071fcf51ebe9dc0cf7 125ca8f4dc853060ee8821d443134fb2618c90d8 M	src/CloudBuildMiner/Program.cs
59	3	src/CloudBuildMiner/Configuration/Configuration.cs
12	4	src/CloudBuildMiner/Program.cs
```

DB Schema

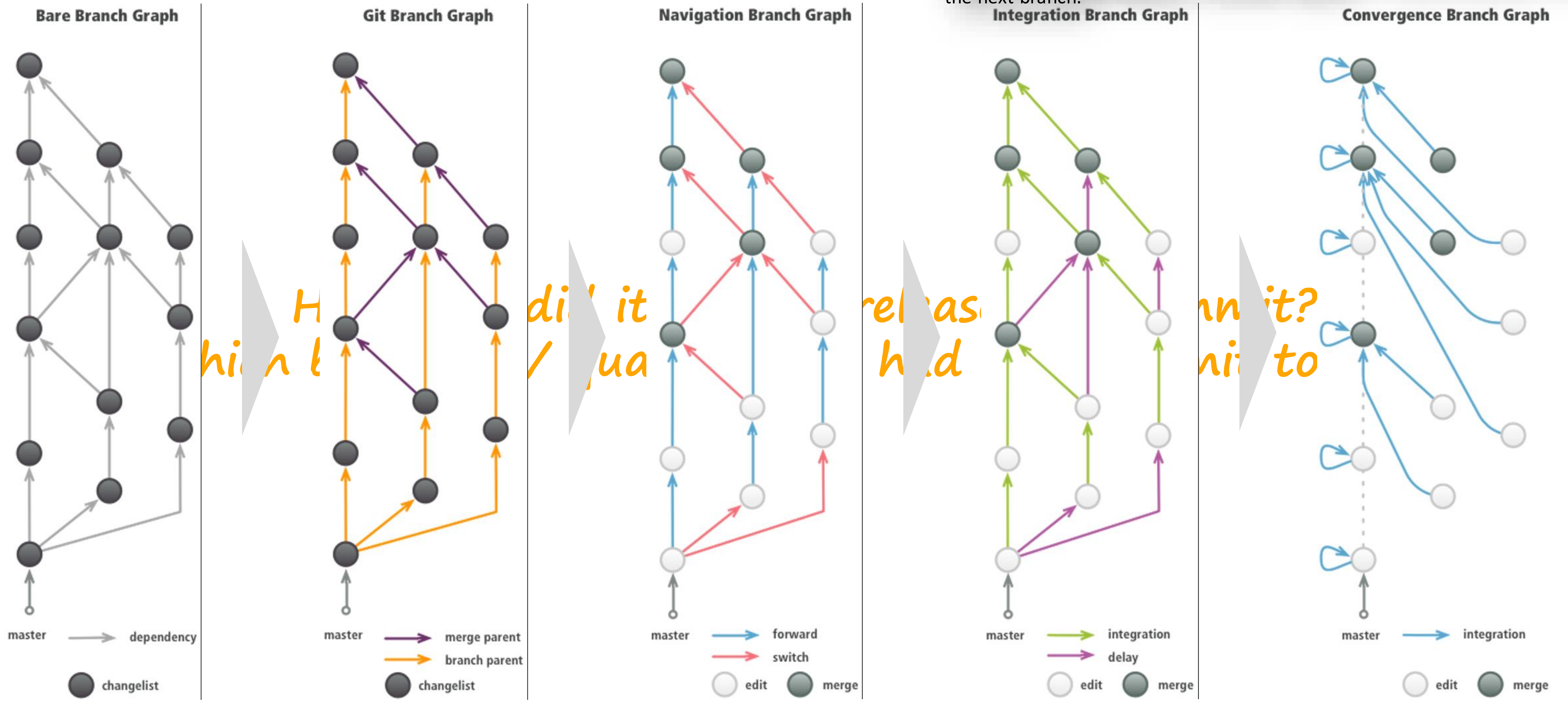

Integrations
tx_id_to
bp_id_to
CodePathFileChanges
fc_id
tx_id
bp_id
rp_id
depot_id
cl_number
cl_time
actype_flags
chg_id
FileChanges
fc_id
tx_id
bp_id
rp_id
depot_id
cl_number
cl_time
actype_flags
chg_id
int_id
undo
kloc_lines
kloc_lines_code
diff_lines_added
diff_lines_deleted
diff_lines_changed_prev
diff_lines_changed_cur
ChangeTransactions
tx_id
user_id
cl_comment
tx_first_time
tx_last_time
hashed_comment
RelativeFilePaths
rp_id
depot_id
rp_file_path
rp_file
rp_ext
rp_local_file_path
ChangeTransact
tx_id
depot_id
cl_number
ChangeTransactions
tx_id
user_id
cl_comment
tx_first_time
tx_last_time
hashed_comment
Users
user_id
user_name
sysac
ChangeTransactionsToChangeLists
tx_id
depot_id
cl_number
FileChanges
fc_id
tx_id
bp_id
rp_id
depot_id
cl_number
cl_time
actype_flags
chg_id

Advanced Mining e.g. code integration paths

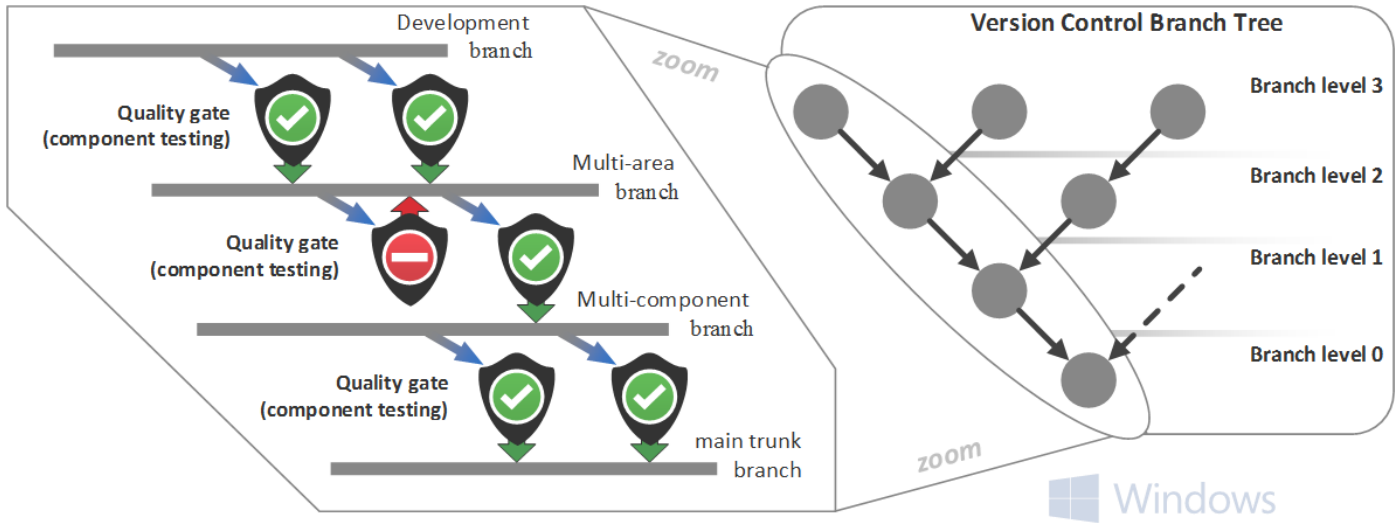
Finding shortest paths

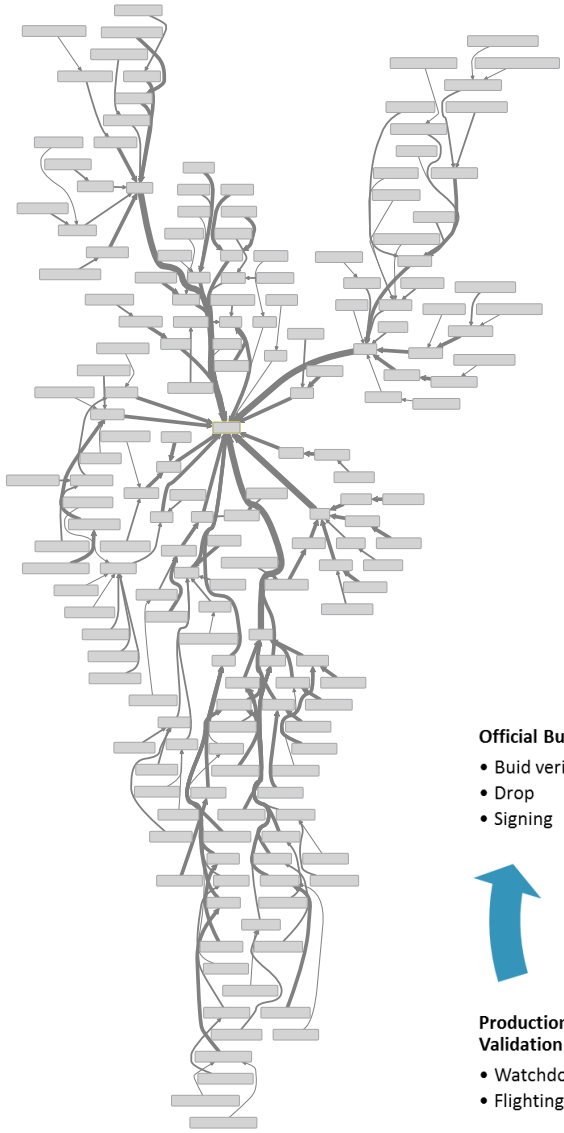
Thanks to Sascha Just (just@st.cs.uni-saarland.de)

the time the commit spends in a branch before it is integrated into the next branch.



Branch structures can be complex …

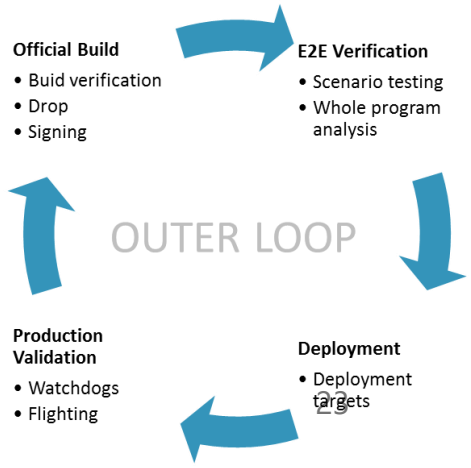




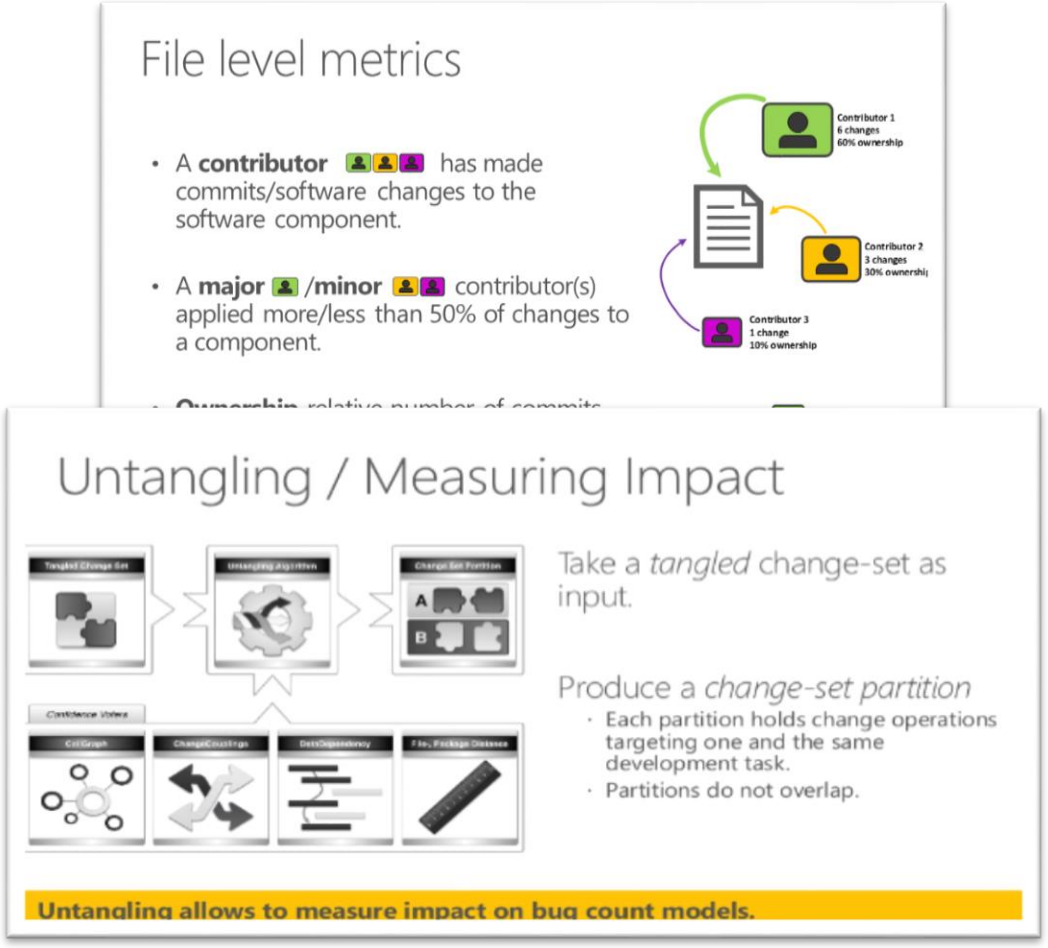
Software testing is expensive

- 10k+ gates executed, 1M+ test cases
- Different branches, architectures, languages, devices, platforms, …

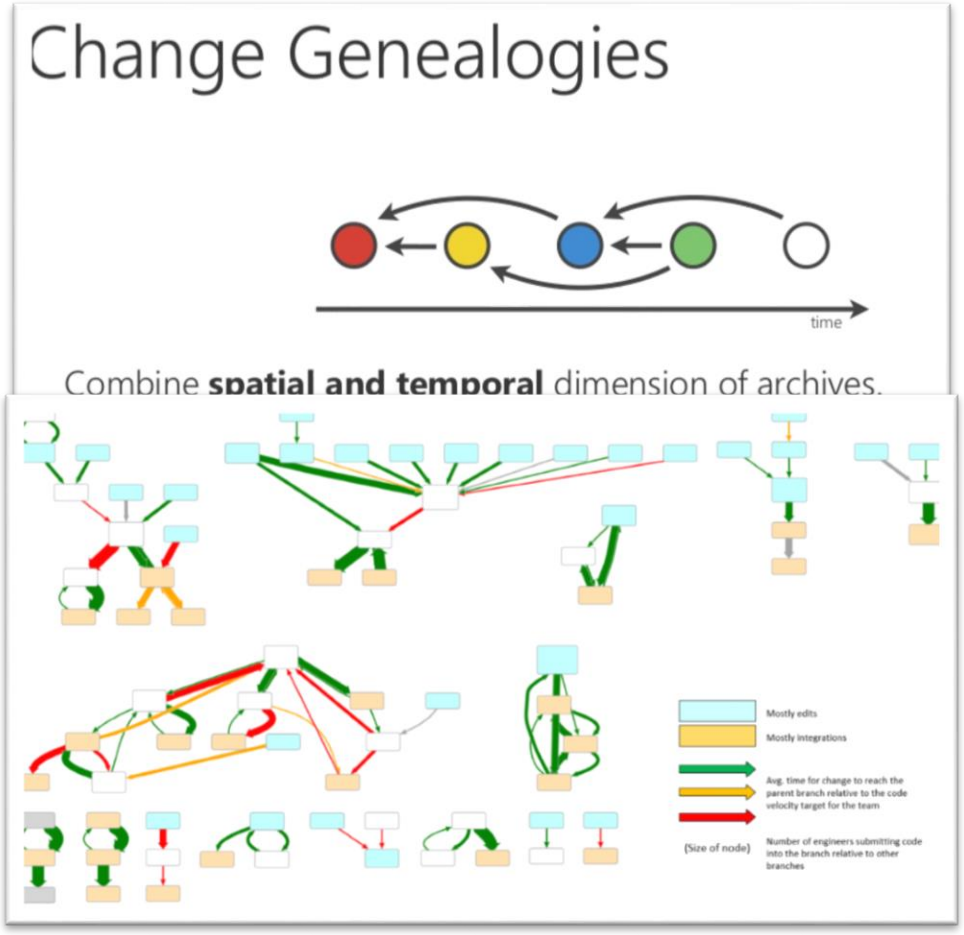
- Aims to find code issues as early as possible
- Slows down product development



Data Usage Scenarios



"The Impact of Tangled Code Changes", Herzig, Zeller, MSR 2013



Code integration paths, Murphy, Czerwonka

Git is "lying" *a matter of perspective*

Fast-Forwards

Issue: original branch info lost
Impact: e.g. shared branches
Prevent: collect meta-data

Rebase

Issue: actual patch changes
Impact: churn metrics, e.g. code quality
Prevent: ban rebases between branches

Apply

Issue: original info lost, e.g. timestamp & author
Impact: valuable data loss
Prevent: `format-patch` & `am`

Squash

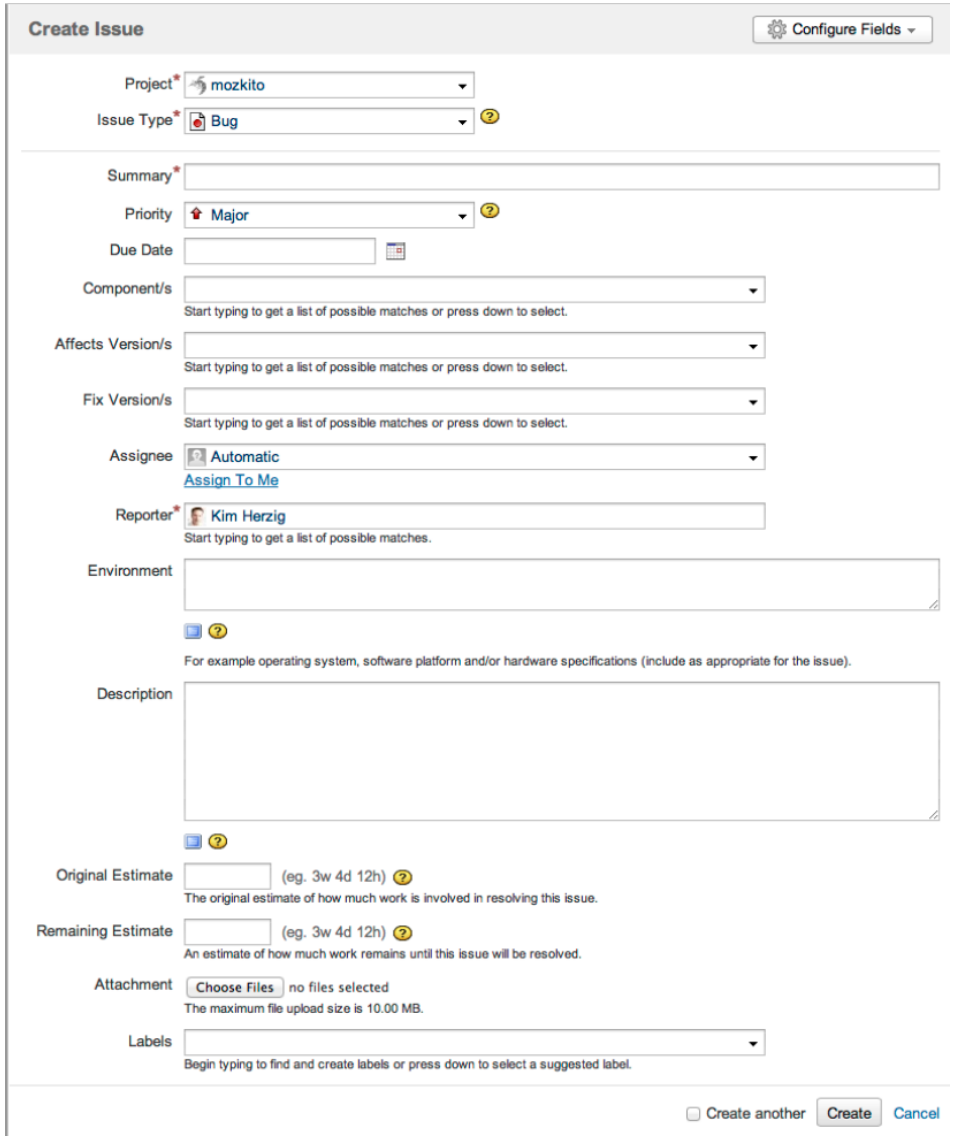
Issue: separate commits tangled
Impact: tangled changes, e.g. code quality
Prevent: ban squash commits

2

It’s not a bug it’s a feature, isn’t it?

Issue Reports

The Basics



Details

Type:	Bug	Status:	Closed (View Workflow)
Priority:	Critical	Resolution:	Fixed
Affects Version/s:	04-SNAPSHOT	Fix Version/s:	None
Component/s:	mozkito-persistence		
Labels:	None		

Description

Activity

All | Comments | Work Log | History | Activity | Source | Reviews

Kim Herzig added a comment - 07/Dec/12 10:29 AM please pull revision 636b97f24b539c74f880e17205d26906b5baa7e6 to reproduce

Kim Herzig added a comment - 07/Dec/12 12:04 PM
Same holds for RepositorySettings. Or I don't get it. A test on GIT runs. A test on HG fails.

Sascha Just added a comment - 08/Dec/12 12:39 PM
Well, this is what you get eventually when using Strings as configuration options...
This was a mess from the beginning and originates from the days we ported the hibernate code.

Interesting fact: This bug will be solved using refactoring Now classify this my friend!

Steps to solve this issue (in progress)

- Replace all occurrences of String databaseType by DatabaseType databaseType.
- Overwrite the toString() in DatabaseType to return name().toLowercase()
- Add constructor to DatabaseType(String) that takes the driver string as an argument
- Add .getDriver() to DatabaseType that returns this string.
- Remove all occurrences of DatabaseType.name() in the code.

Sascha Just added a comment - 08/Dec/12 12:43 PM I also added DatabaseType.available() to check if we have the driver class on the classpath.

Sascha Just added a comment - 08/Dec/12 12:48 PM Out of context: I added 2 static methods to UnrecoverableError: public static UnrecoverableError forma

Sascha Just added a comment - 08/Dec/12 12:52 PM To avoid messy code, I removed the following method from the PersistenceUtil interface: void createSe

Sascha Just added a comment - 08/Dec/12 1:33 PM Additionally introducing a new class to wrap around database options. Validity checks are scattered acr

Sascha Just added a comment - 08/Dec/12 1:37 PM Further, I rename the values in ConnectOptions. These do not reflect the actual actions caused anymor

Sascha Just added a comment - 08/Dec/12 3:09 PM Additionally: removed driver option from DatabaseTest annotation and replaced type string by enum.

Sascha Just added a comment - 08/Dec/12 4:12 PM The reported bug will be fixed after the next push. However, there were more problems with the actual t

Sascha Just <sascha.just@own-hero.net> submitted changeset f4f9b83f9856cb955debc8657448a34f7b950f04 to master in mozkito-research (15 files) - 08/De

Sascha Just made changes - Saturday 4:24 PM

Field	Original Value	New Value
Status	Open [1]	Closed [6]
Resolution		Fixed [1]

API versus HTML

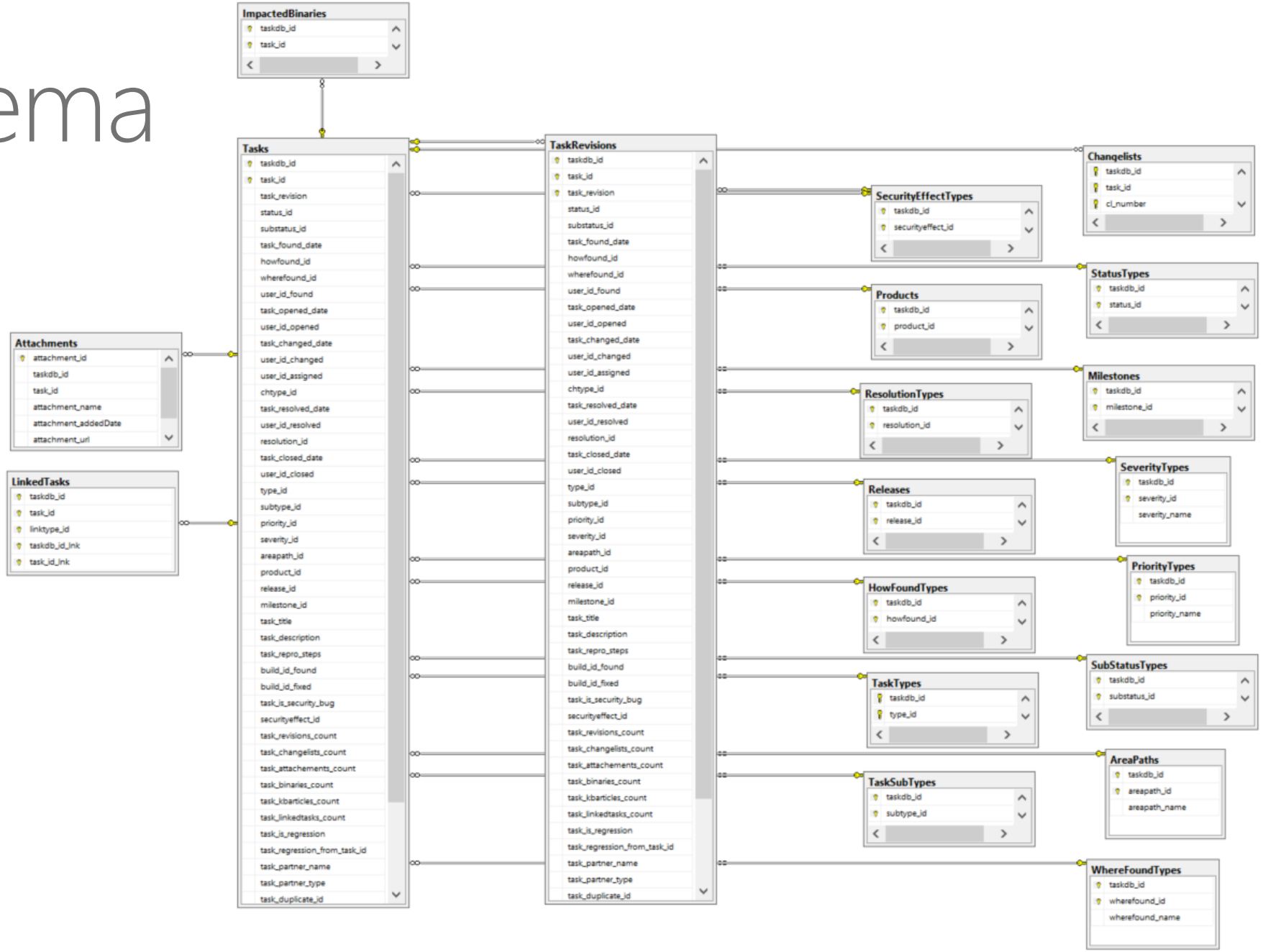
Using the bug tracker API

- Easier
- Often missing info e.g. history
- APIs tend to change often

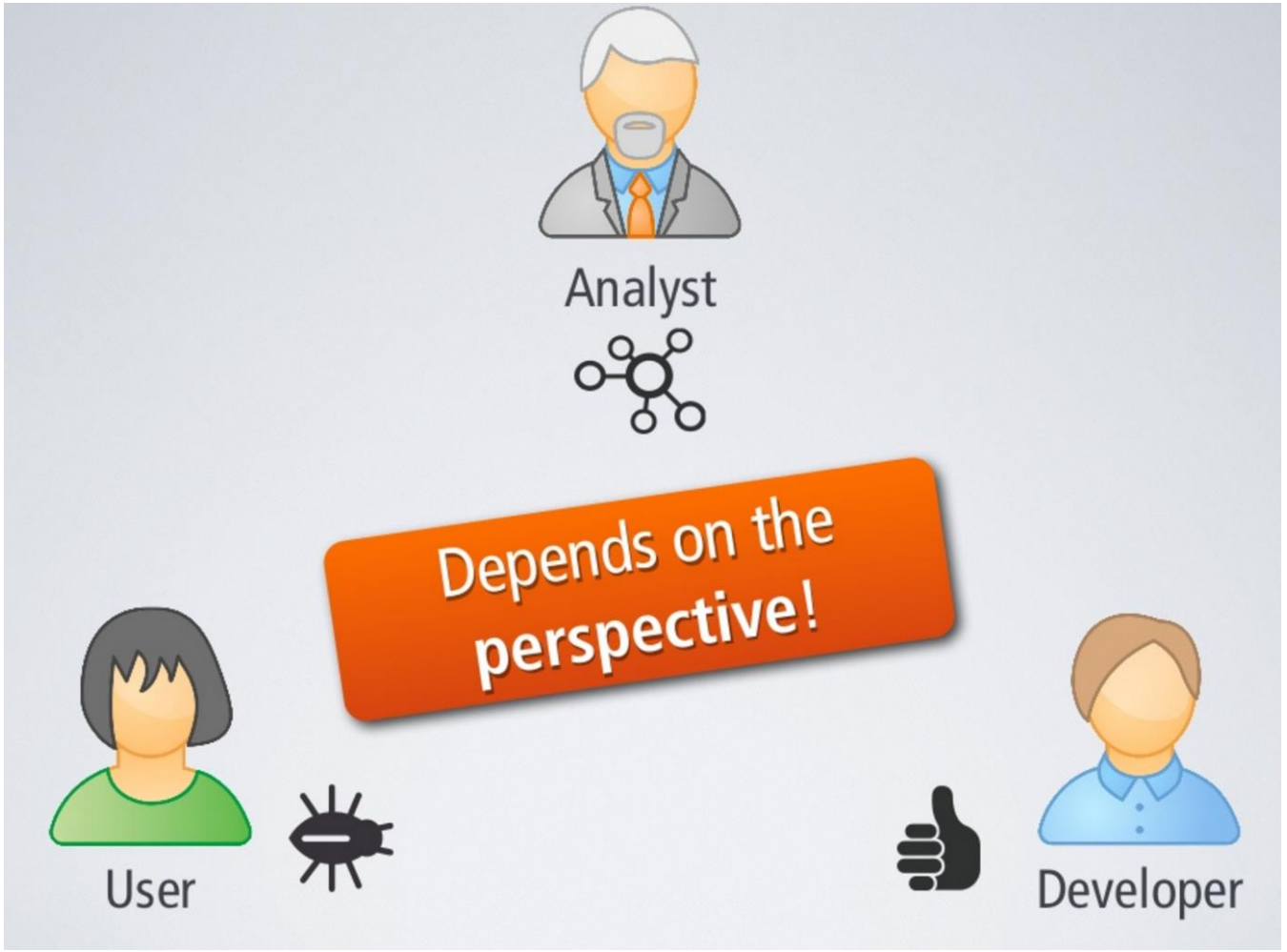
Parse plain HTML / XML

- Access to all info
- More effort
- Will break often. Requires good testing / automation.

DB Schema

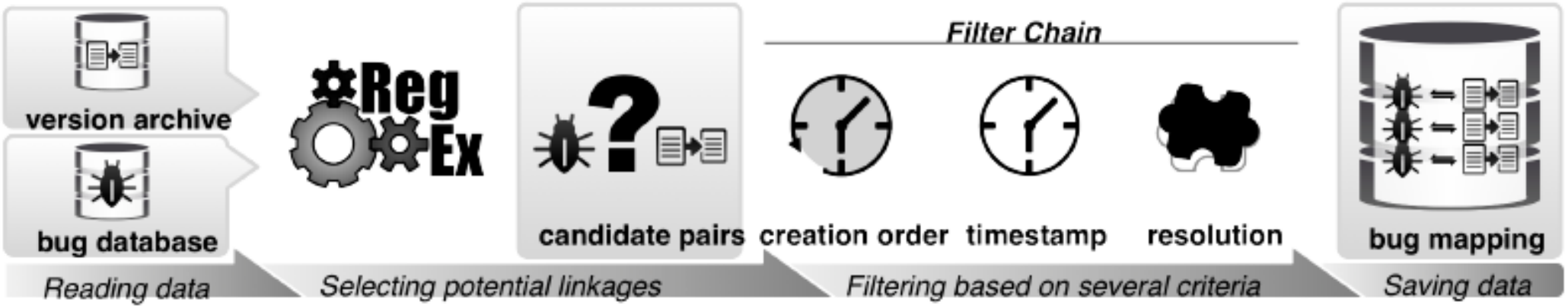

ImpactedBinaries
taskdb_id
task_id
Tasks
taskdb_id
task_id
task_revision
status_id
substatus_id
task_found_date
howfound_id
wherefound_id
user_id_found
task_opened_date
user_id_opened
task_changed_date
user_id_changed
user_id_assigned
chtype_id
task_resolved_date
user_id_resolved
resolution_id
task_closed_date
user_id_closed
type_id
subtype_id
priority_id
severity_id
areapath_id
product_id
release_id
milestone_id
task_title
task_description
task_repro_steps
build_id_found
build_id_fixed
task_is_security_bug
securityeffect_id
task_revisions_count
task_changelists_count
task_attachements_count
task_binaries_count
task_kbarticles_count
task_linkedtasks_count
task_is_regression
task_regression_from_task_id
task_partner_name
task_partner_type
task_duplicate_id
Attachments
attachment_id
taskdb_id
task_id
attachment_name
attachment_addedDate
attachment_url
LinkedTasks
taskdb_id
task_id
linktype_id
taskdb_id_lnk
task_id_lnk
TaskRevisions
taskdb_id
task_id
task_revision
status_id
substatus_id
task_found_date
howfound_id
wherefound_id
user_id_found
task_opened_date
user_id_opened
task_changed_date
user_id_changed
user_id_assigned
chtype_id
task_resolved_date
user_id_resolved
resolution_id
task_closed_date
user_id_closed
type_id
subtype_id
priority_id
severity_id
areapath_id
product_id
release_id
milestone_id
task_title
task_description
task_repro_steps
build_id_found
build_id_fixed
task_is_security_bug
securityeffect_id
task_revisions_count
task_changelists_count
task_attachements_count
task_binaries_count
task_kbarticles_count
task_linkedtasks_count
task_is_regression
task_regression_from_task_id
task_partner_name
task_partner_type
task_duplicate_id
SecurityEffectTypes
taskdb_id
securityeffect_id
Products
taskdb_id
product_id
ResolutionTypes
taskdb_id
resolution_id
Releases
taskdb_id
release_id
HowFoundTypes
taskdb_id
howfound_id
TaskTypes
taskdb_id
type_id
TaskSubTypes
taskdb_id
subtype_id
Changelists
taskdb_id
task_id
cl_number
StatusTypes
taskdb_id
status_id
Milestones
taskdb_id
milestone_id
SeverityTypes
taskdb_id
severity_id
severity_name
PriorityTypes
taskdb_id
priority_id
priority_name
SubStatusTypes
taskdb_id
substatus_id
AreaPaths
taskdb_id
areapath_id
areapath_name
WhereFoundTypes
taskdb_id
wherefound_id
wherefound_name

The problem with issue reports is not the mining part, It's the data interpretation part. :-P

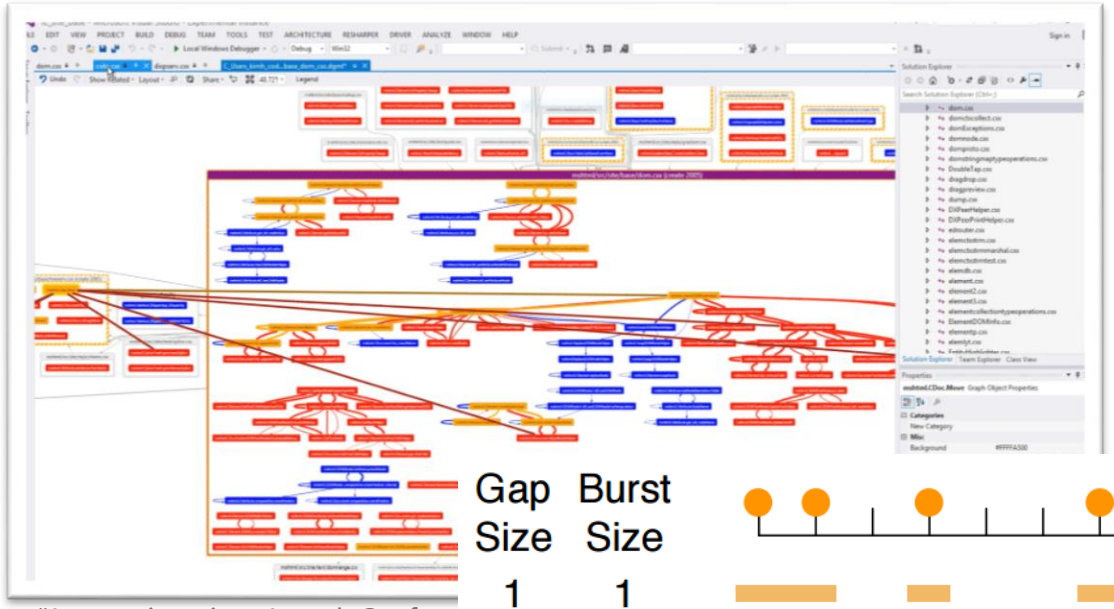


Mapping bugs to code

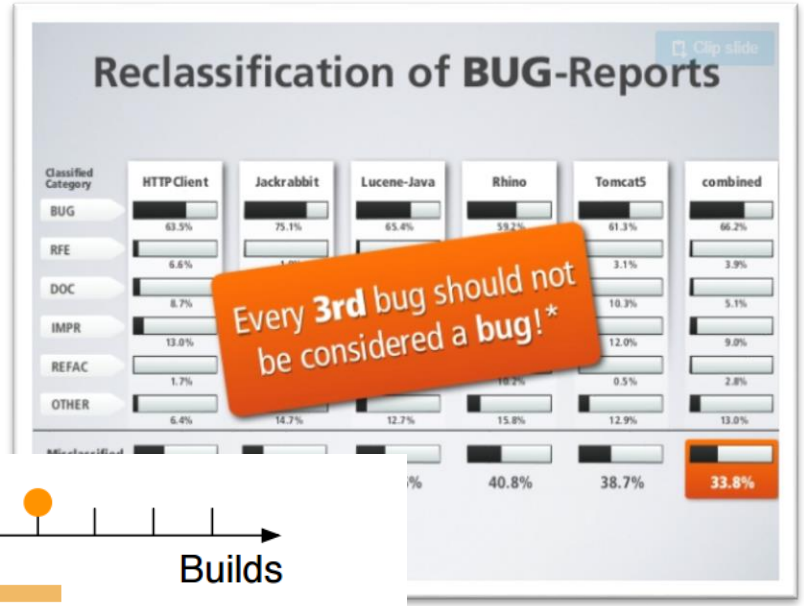
- Check for completeness / correctness of keywords constantly
  - TIP: Monitor mapping rate. Any drop is likely a due to changes in patterns / templates.
- Timestamps are always a challenge.
  - Bugs might be closed after commits (opened after the fact)
  - Different server in different time zones, …
- Not all bugs get closed
  - There might be hidden patterns for “closed” reports



Data Usage Scenarios



"Approximating Attack Surface
Morrison, Murphy, Williams, IC



ıre — How
Bug Prediction",
015

"Change Bursts as Defect Predictors", Nagappan, Zeller, Zimmermann, Herzig, Murphy
ISSRE 2010

Issue Report Pitfalls

- Different tracker => different behavior
  - **Tracker represent data differently**. Users will misuse fields or represent
- Default values
  - **Many issue reporters use default values**: they might not know the righ
- What is a bug and what an improvement
  - **Philosophical question!** Be aware that a bug might not be a bug for everyone
- Ambiguous terms
  - **Likely to introduce noise**.
- No standard field set
  - **Labels or tags may replace fields**, e.g. Google tracker or Sourceforge
  - Hard to interpret as there is no common behavior , not even within a te

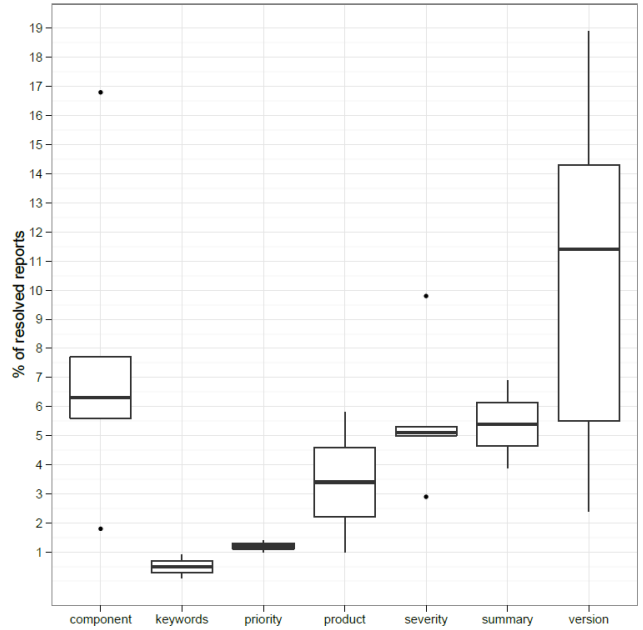
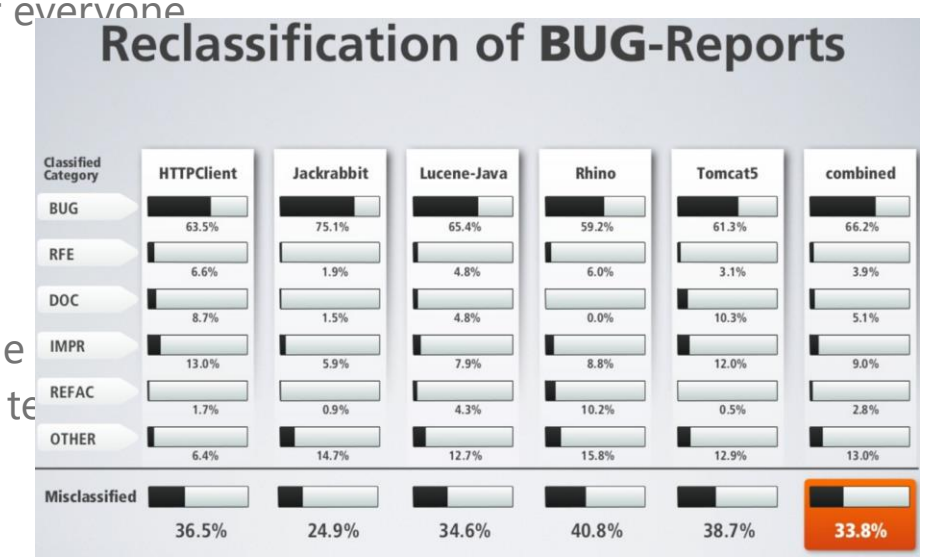


Fig. 2: Percent of resolved issue reports with respect to field changes. Priority, product, and summary only changed in BUGZILLA tracker projects RHINO and TOMCAT.



3

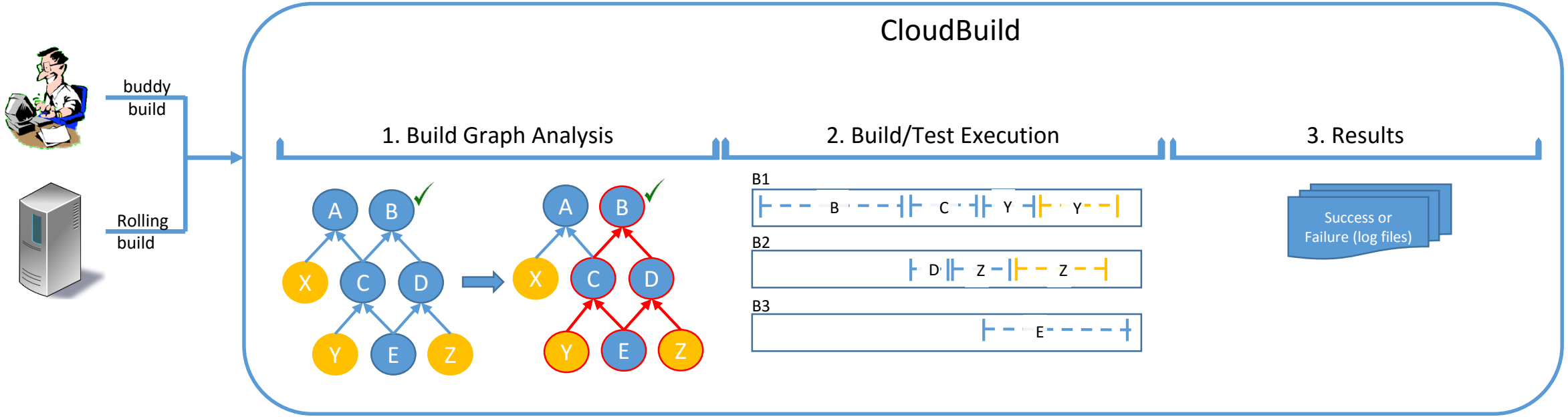
Who cares about build? You should!

Build Data

Why Build Data?

- **Lower bound of how fast we can ship software**
- Compiling complex software can take hours.
  - Very expensive!
- Speedup may require componentization / refactoring.
  - Very expensive!
- Catch 22: faster development usually means more builds.
- Each build **failure brings the development pipeline to halt**.
- Modern build systems are extremely complex!

CloudBuild @ Microsoft



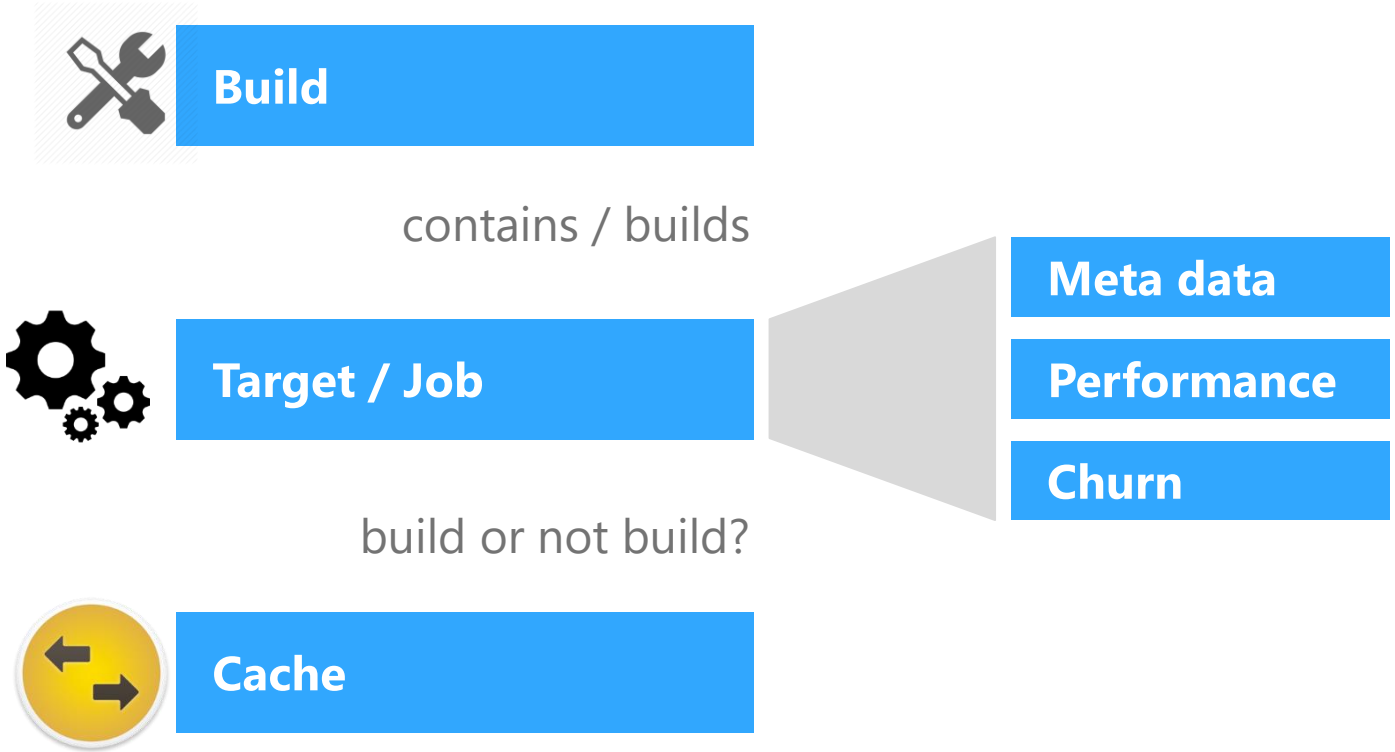
Legend:
- Source module
- Test module
- Build
- Test execution

CloudBuild uses module-level test selection

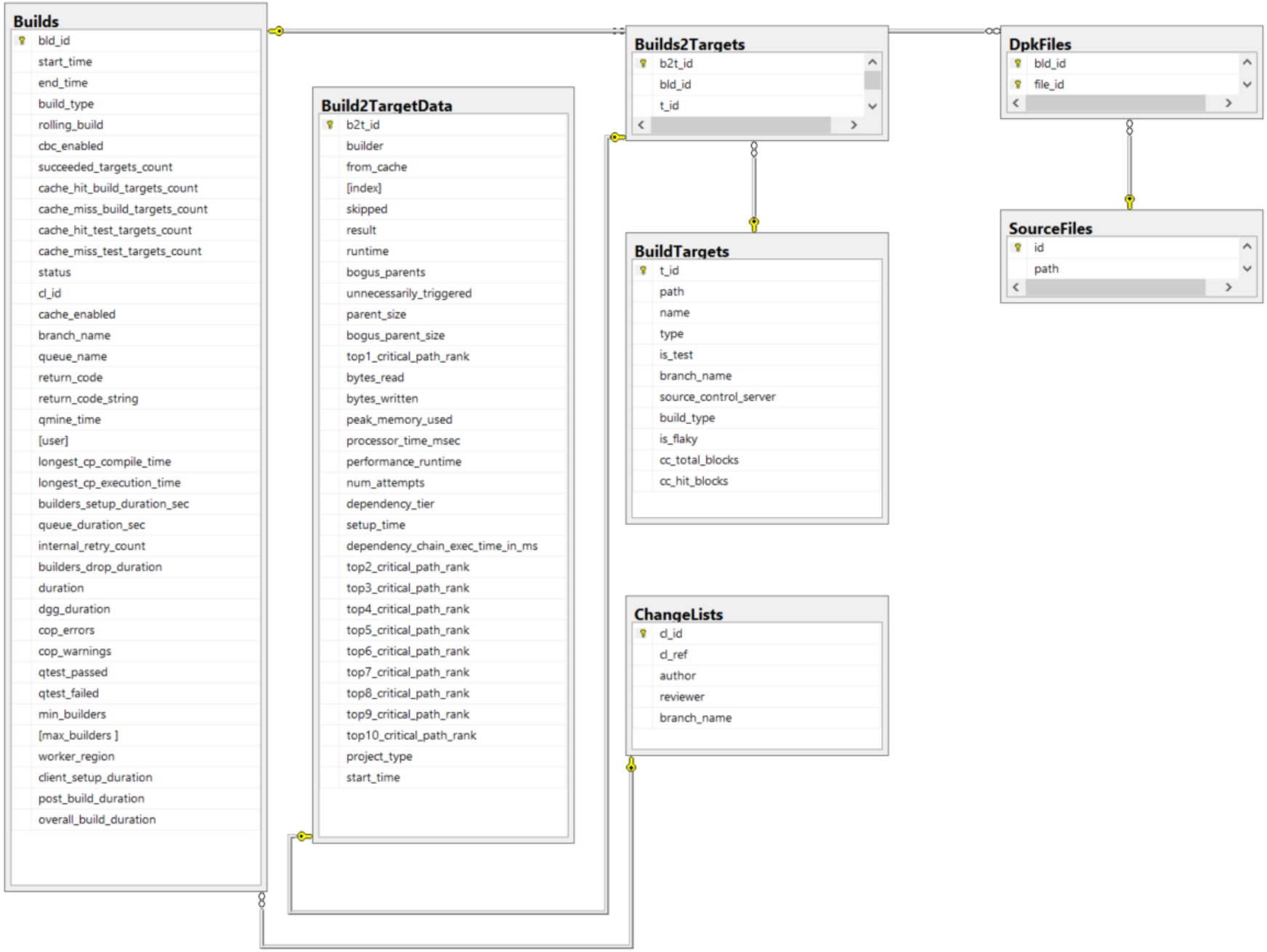
when any dependencies of a module change, all tests inside that module are executed.

The Basics

Many aspects will rely on build telemetry data (white box)

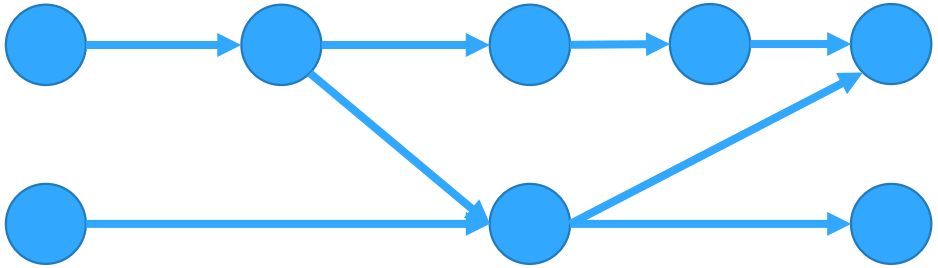


DB Schema

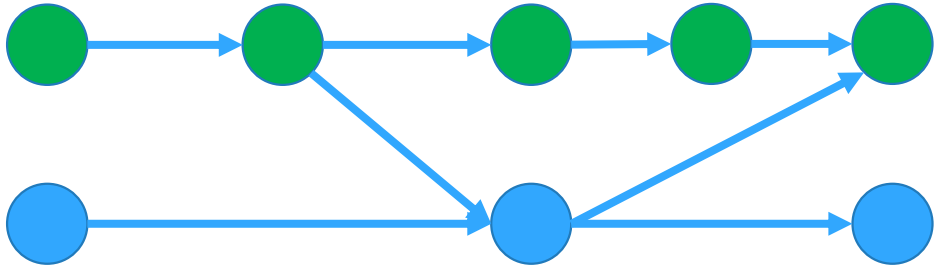

Builds
bld_id
start_time
end_time
build_type
rolling_build
cbc_enabled
succeeded_targets_count
cache_hit_build_targets_count
cache_miss_build_targets_count
cache_hit_test_targets_count
cache_miss_test_targets_count
status
cl_id
cache_enabled
branch_name
queue_name
return_code
return_code_string
qmine_time
[user]
longest_cp_compile_time
longest_cp_execution_time
builders_setup_duration_sec
queue_duration_sec
internal_retry_count
builders_drop_duration
duration
dgg_duration
cop_errors
cop_warnings
qtest_passed
qtest_failed
min_builders
[max_builders]
worker_region
client_setup_duration
post_build_duration
overall_build_duration
Build2TargetData
b2t_id
builder
from_cache
[index]
skipped
result
runtime
bogus_parents
unnecessarily_triggered
parent_size
bogus_parent_size
top1_critical_path_rank
bytes_read
bytes_written
peak_memory_used
processor_time_msec
performance_runtime
num_attempts
dependency_tier
setup_time
dependency_chain_exec_time_in_ms
top2_critical_path_rank
top3_critical_path_rank
top4_critical_path_rank
top5_critical_path_rank
top6_critical_path_rank
top7_critical_path_rank
top8_critical_path_rank
top9_critical_path_rank
top10_critical_path_rank
project_type
start_time
Builds2Targets
b2t_id
bld_id
t_id
BuildTargets
t_id
path
name
type
is_test
branch_name
source_control_server
build_type
is_flaky
cc_total_blocks
cc_hit_blocks
ChangeLists
cl_id
cl_ref
author
reviewer
branch_name
DpkFiles
bld_id
file_id
SourceFiles
id
path

Longest Critical Build Path (LCP)

- Graph structure: Directed Acyclic Graph (DAG)
  - Edges define dependencies.

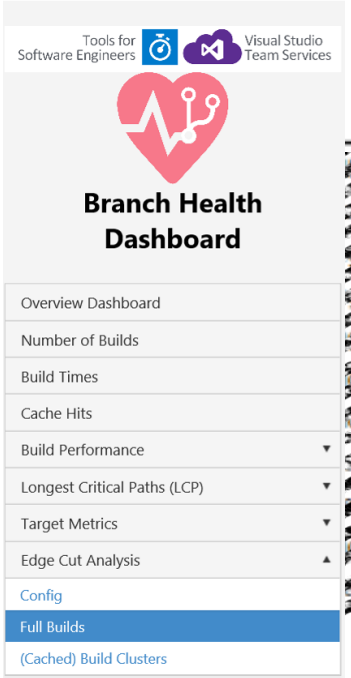


- LCP: *Defines the* ***lower bound on how fast we can build*** *the DAG*
  - Defined with respect to build time, not target count.



You will need to break the LCP to improve build times!

Data Usage Scenarios



- Suggesting refactorings to cut build time (*what-if-analysis*)

Parent	Child	#Calls	%Speedup	%Accumulated Speedup	Black-list
[redacted]	[redacted]	1	0.00 %	0.00 %	☐
		1	4.64 %	4.64 %	☐
		2	0.02 %	4.67 %	☐
		2	0.00 %	4.80 %	☐
		2	0.00 %	4.80 %	☐
		2	0.02 %	4.82 %	☐
		2	0.05 %	4.85 %	☐
		2	0.03 %	4.88 %	☐
		2	0.00 %	4.90 %	☐
		3	0.20 %	4.93 %	☐
		4	1.39 %	6.32 %	☐

- Predict impact of churn on build times.

FilePath	#CLs	Trend	CL IDs	#Targets churned	#Targets built
[redacted]	7		2006795, 2008798, 2016623, 2016702, 2017669, 2017892, 2020499	5	868

-> Great for policy making, e.g. pre-checking

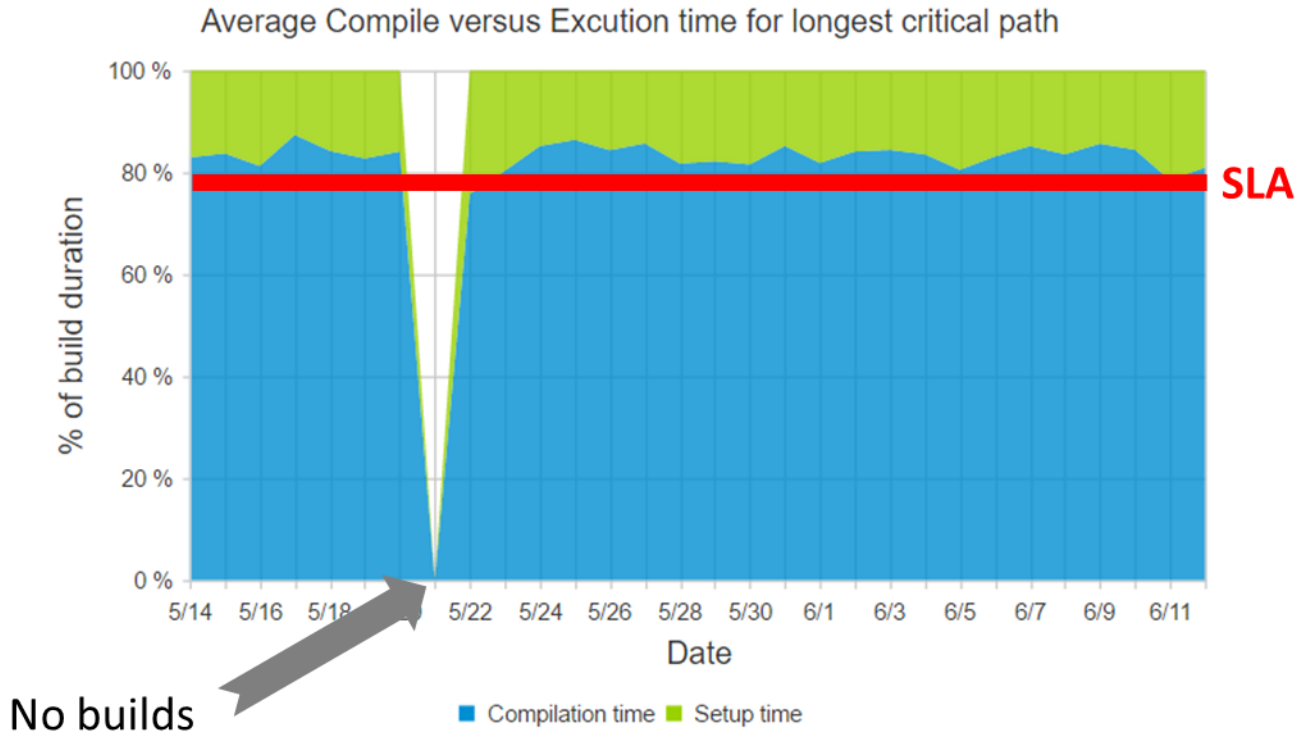
Data Usage Scenarios (2)

- Detecting architectural build bottlenecks

Top 25 targets most frequently appearing on longest critical path.

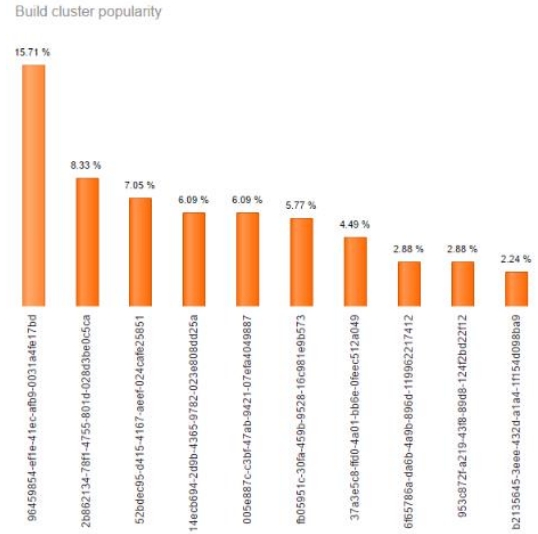
Target Name	%Builds
[illegible]	91.58 %
[illegible]	90.84 %
[illegible]	88.37 %
[illegible]	88.12 %
[illegible]	87.87 %
[illegible]	87.62 %
[illegible]	84.16 %

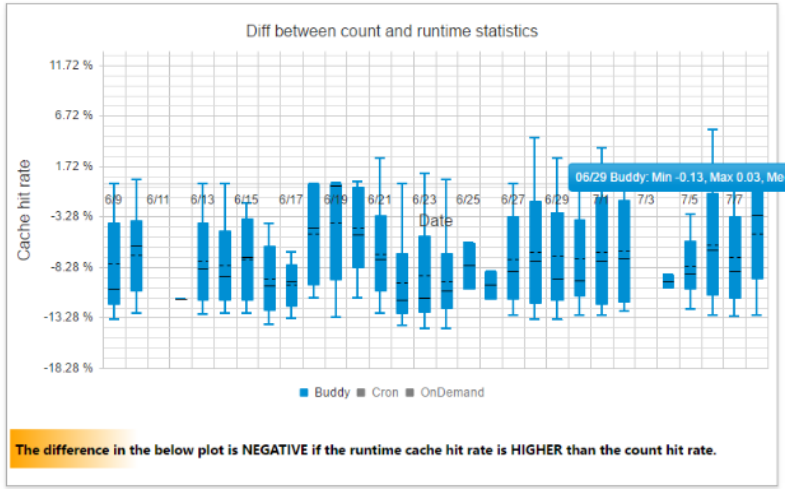
- Verifying build tool SLAs



Common Build Data Pitfalls

- No two builds are the same (caching)
  - **Cluster builds based on similarity.** Definition of similarity depends on scenario.
- Only compare comparable builds
  - **Cache hit rate, passing or failing, same code base, build type, etc.**
- Cache hit rates can be misleading
  - **Counts versus build time.**
- Build performance has many factors
  - **Churn, machine capacity, size, I/O, time window, caching, network, …**





4

Where research and industry drift apart

Test Executions

The Basics

Very expensive to mine after the fact. Collect telemetry!

- **Store all results of all test runs! 24/7 - 365**
  - Data can become large. SQL is no option.
- Record **associations** with builds / users / time / code base
  - Comes in nearly all cases for free.
- Record **details for failures**, e.g. stack traces
- Record **code coverage** *if possible*
  - HUGE data requirements
- Record **performance info**:
  - How long did the test run? CPU usage.

~~DB Schema~~ Azure Data Lake

Test target (suite)

Column Name	Type
UniqueSessionId	string
FullBranchName	
Queue	
TargetId	
Result	
Path	
TestRunTimeMS	long
InfrastructureTimeMs	long
Passed	int
Skipped	int
FilesDeployedBytes	long
TestSourceDir	string
TestAdapter	string
DotNetFramework	string
Platform	string
QTestAccountMode	string

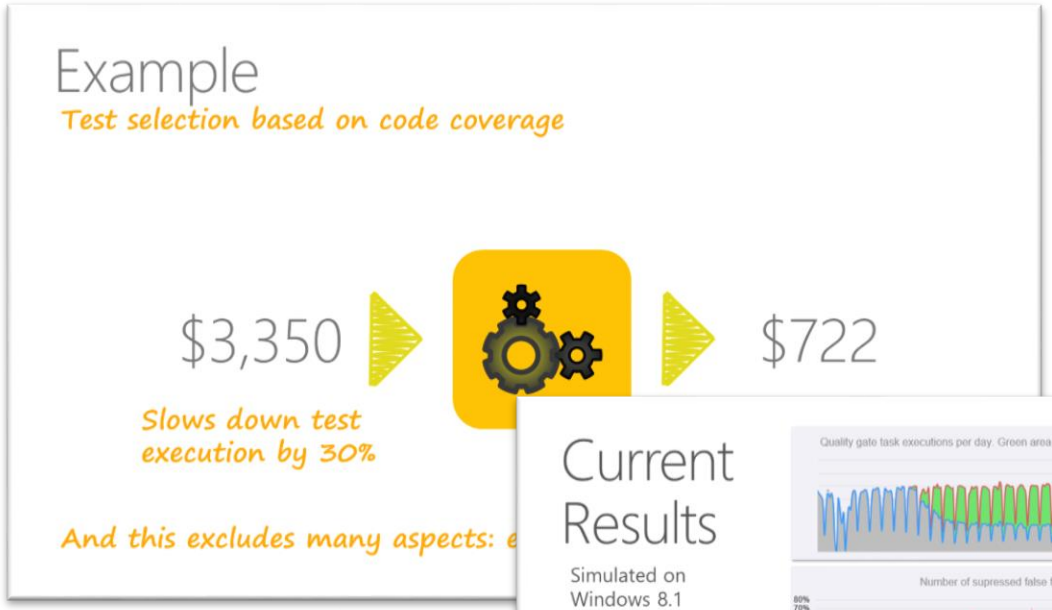
> 1 million test case executions per day per product.
>
> ~ 10GB per day

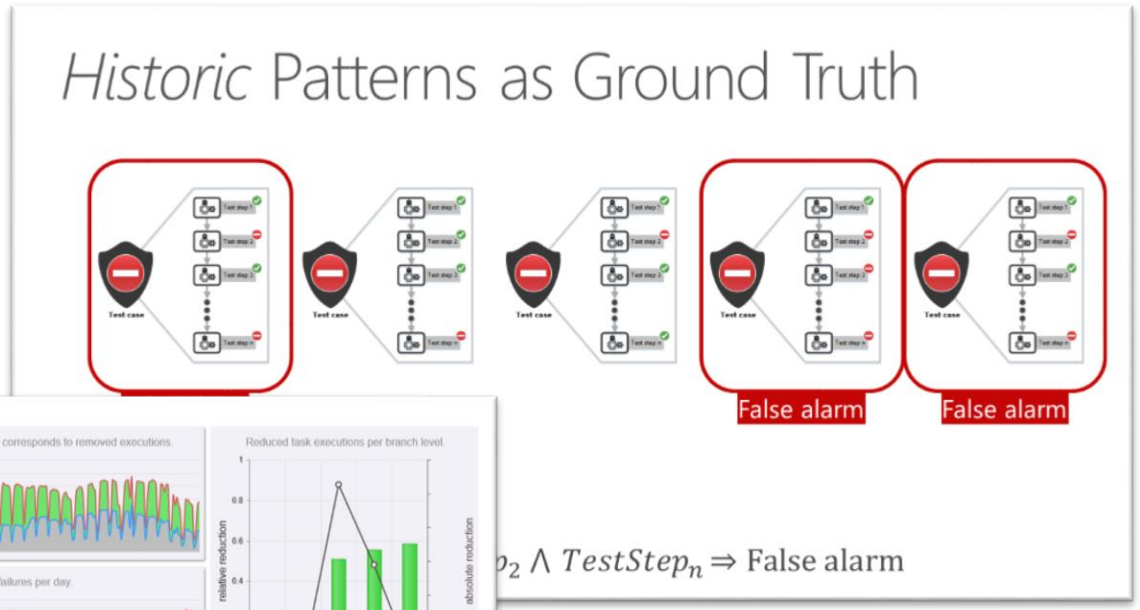
Test case

Schema

Column Name	Type
UniqueSessionId	string
FullBranchName	string
	string
	string
	string
Result	string
Attempts	int
ExecutionTimeMs	float
Flaky	string
Message	string
StackTrace	string

Data Usage Scenarios





"Let's assume we had to pay for te

Alarms Using Association Rules", Herzig,

"The Art of Testing Less Without Sacrificing Code Quality", Herzig, Greiler, Czerwonka, Murphy, ICSE 2015

Common Test Data Pitfalls

- **Flaky tests**: not every failure is a bug.
  - In fact, most failures are false test alarms.
  - Is it a flaky passing or flaky failing tests?
  - Flaky tests might get re-run.
- **Nobody runs all tests** anymore.
  - Every test event gives you an “incomplete” picture.
- Tests are run in **parallel with build**.
  - Saving test runs may be not as beneficial as you might think.
- Consider the **history** and meta data.
  - Owner, purpose, age, failure rate, run time, expense of a failure, who/how triggered the test …
- **Code base is moving**. No two test runs are the same.
  - Except if you specifically control for it.

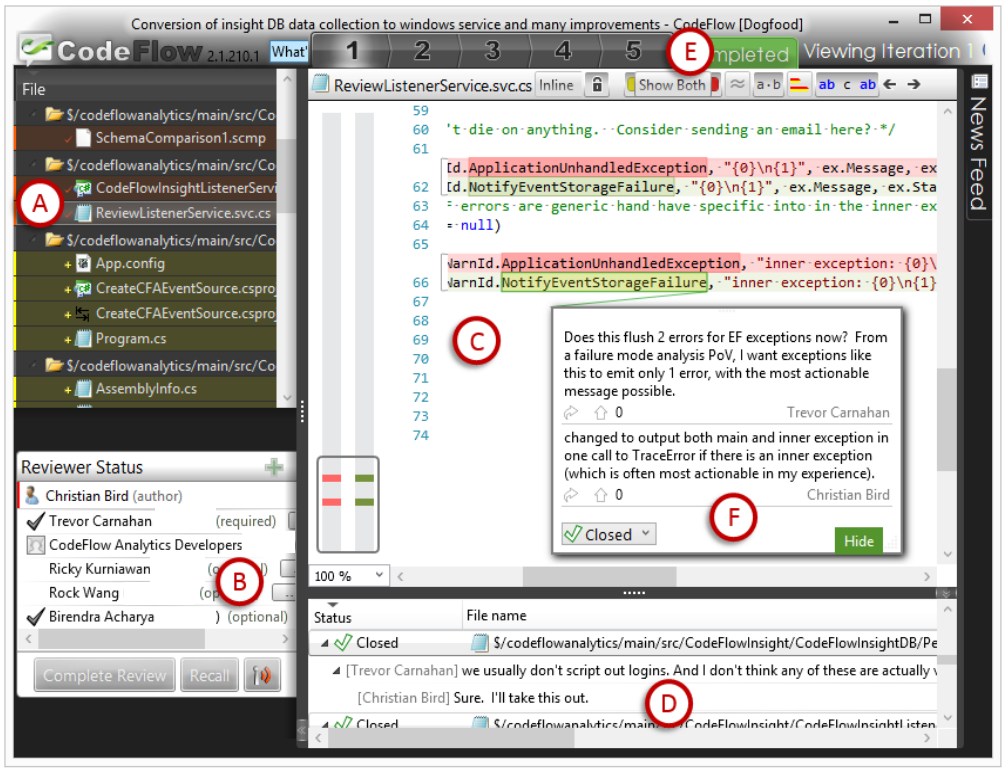
5

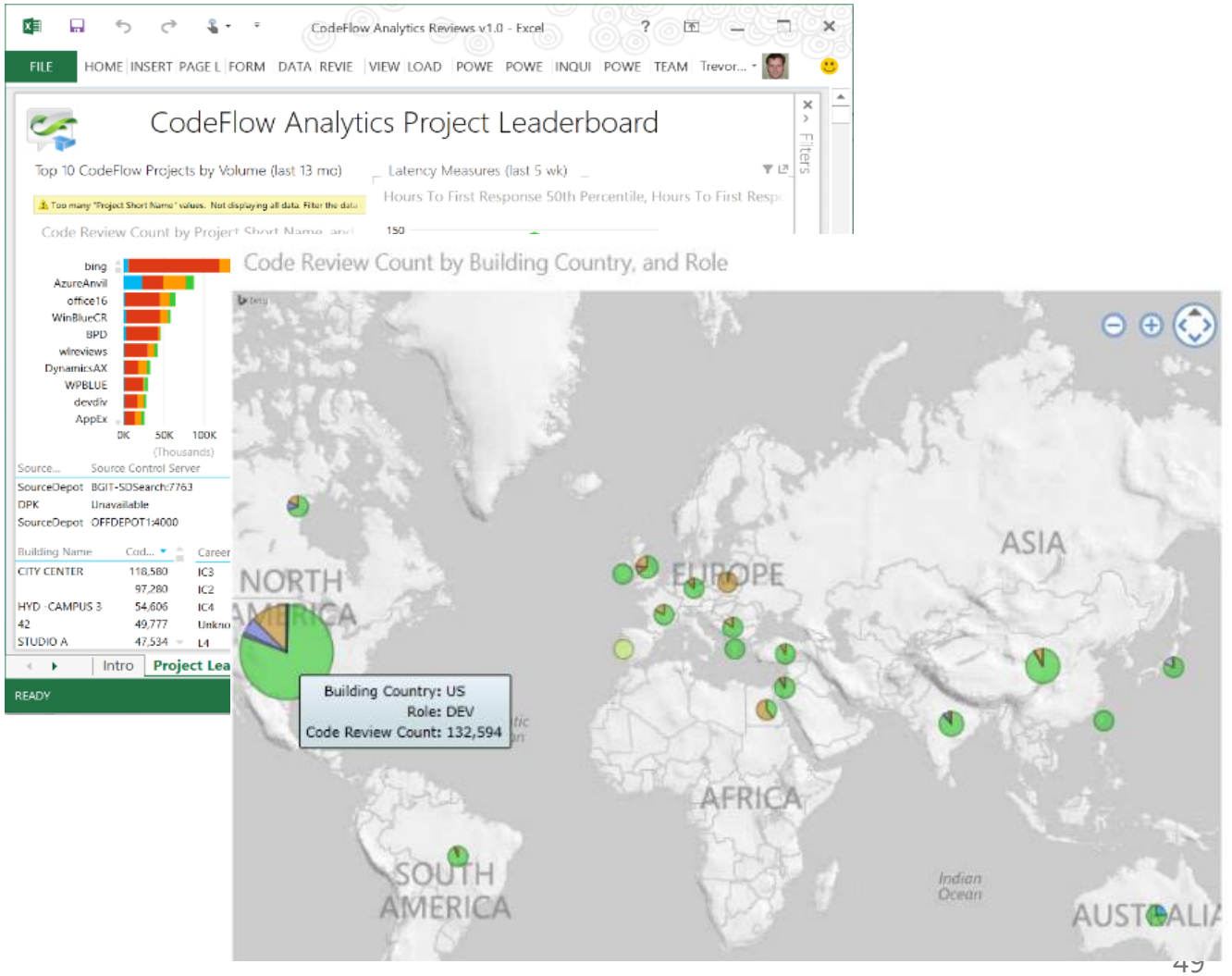
"En vogue" but misunderstood

Code Reviews

The Basics

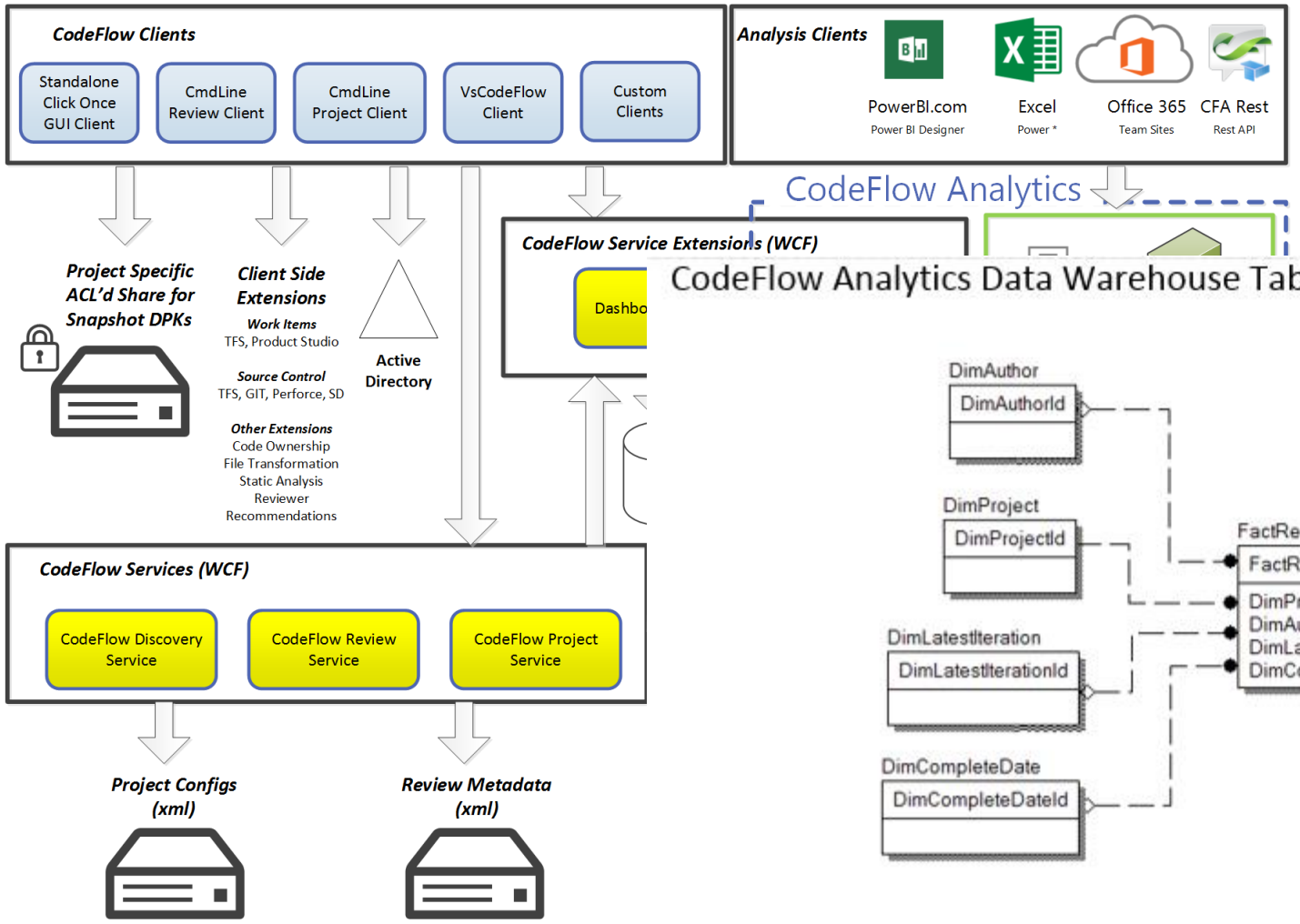
Thanks to Kivanc Muslu <kivancm@microsoft.com>



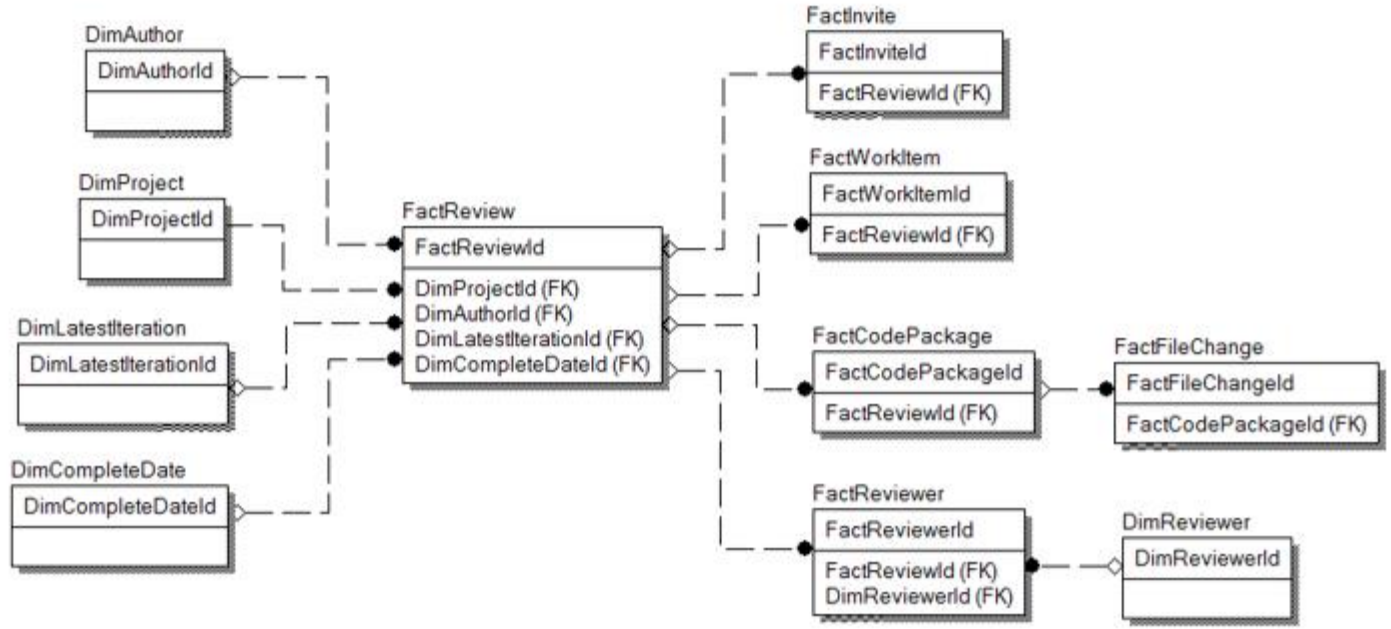


The Basics

Thanks to Kivanc Muslu <kivancm@microsoft.com>



CodeFlow Analytics Data Warehouse Table Relationships:



Where to collect data?

Thanks to Kivanc Muslu <kivancm@microsoft.com>

Server side (service)

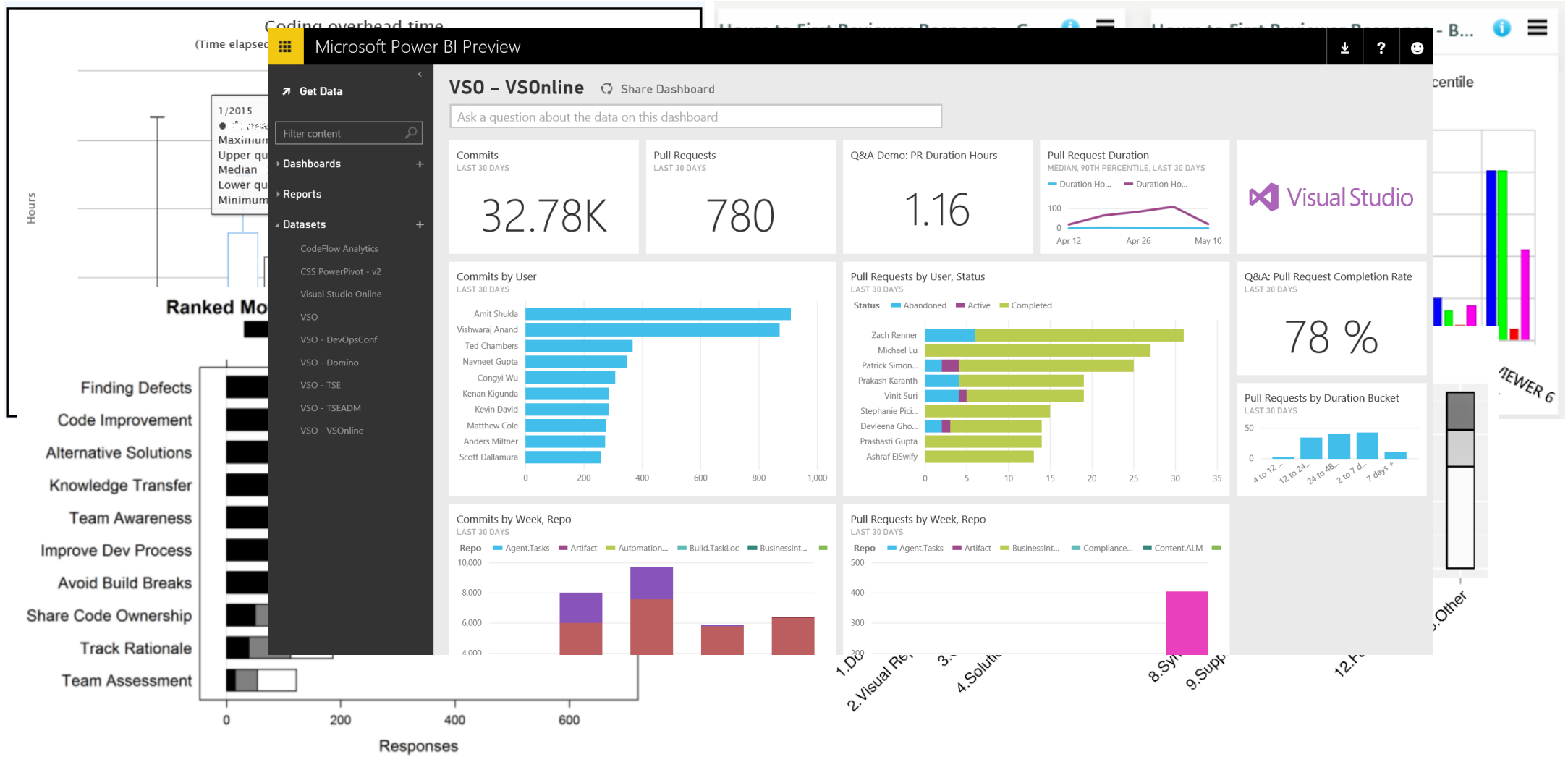
- *Preferred*
- Not all signals/metrics available
- Instant release (no cross-over)
- Full control

Client side

- May come from different client versions
- Making mistakes can be fatal
  - Runtime, hard to revert, long time to fix
- Data loss due to setups
  - E.g. 4500 users on service, 4000 on client
- Difficult to debug data loss or issues

Data Usage Scenarios

Thanks to Kivanc Muslu <kivancm@microsoft.com>



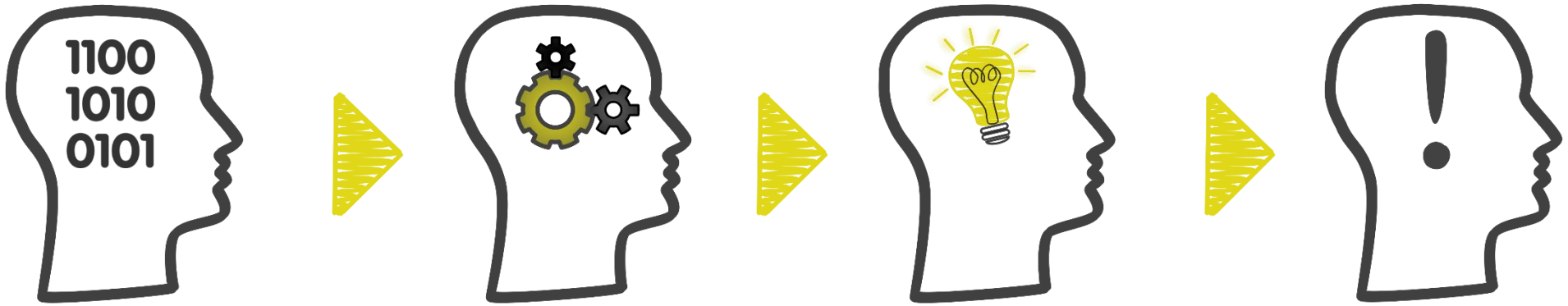
Common Code Review Pitfalls

Thanks to Kivanc Muslu <kivancm@microsoft.com>

- In most cases: **do not use data standalone**.
  - Combing with code, churn, tests, builds, etc.
- Often requires **contextual information**
  - Why are taking the most complex reviews only seconds?
  - Why did the comment exist?
  - Why did no reviewer respond?
- Quality of review depend on **human aspects**
  - E.g. agenda of reviewer, time of submission, engineers block one hour a day to do reviews
- Different **usages** of code reviews
  - Gated check-in
  - FYI
  - Documentation
- **No ground-truth available**

If you take one message from this session …

Mining Software Archives



Tools for Software Engineers

Microsoft