

Code Reviews Do Not Find Bugs

How the Current Code Review Best Practice Slows Us Down

Jacek Czerwonka, Michaela Greiler, Jack Tilford
{ jacekcz, mgreiler, jtilford } @ microsoft.com
Tools for Software Engineers team, Microsoft Corp.

(in collaboration with Christian Bird, Microsoft Research)

File Change

- ✓ /codemine/dev/src
- ✓ CodeMine.sln Edit
- ✓ /codemine/dev/src/Cmc/Database
- ✓ CMC.sqlproj Edit
- ✓ /codemine/dev/src/Cmc/Database/Post-Deplo
- ✓ Loader.Constraints.Data.sql Edit
- ✓ /codemine/dev/src/Cmc/Database/Schemas/dl
- ✓ /codemine/dev/src/Cmc/Database/Schemas/Lc
- ✓ /codemine/dev/src/Cmc/Loader/QBuildLoader
- ✓ /codemine/dev/src/Cmc/Loader/QBuildLoader
- ✓ /codemine/dev/src/Cmc/Loader/QBuildLoader
- ✓ /codemine/dev/src/Cmc/Loader/QBuildLoader
- ✓ /codemine/dev/src/Cmc/Loader/QBuildLoader
- ✓ /codemine/dev/src/Cmc/Loader/QBuildLoader
- ✓ /codemine/dev/src/Common/LoaderBase
- ✓ LoaderBase.csproj Edit
- ✓ LoaderController.cs Edit
- ✓ /codemine/dev/src/Common/LoaderBase/Que
- ✓ QueueManager.cs Edit
- ✓ /codemine/dev/src/Common/LoaderBase/Rest
- ✓ RestCall.cs Add
- ✓ /codemine/dev/src/Test/CodeMine.Integration
- ✓ CodeMine.IntegrationTests.cs Edit
- ✓ /codemine/dev/src/Test/CodeMine.Integration
- ✓ QBuildLoaderSuite.cs Add
- ✓ /codemine/dev/src/Test/CodeMine.Integration
- ✓ qbuild_all.json Add
- ✓ qbuild_finished_1.json Add
- ✓ qbuild_finished_2.json Add

Review Properties

[Predicted risk of defects](#) **Low**

CMC.sqlproj

You must publish your review before other reviewers will be able to see it. Publish... Start Over

```

305     </Build>
306     <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Stored Procedures\GetInvalidConstraints.proc.sql">
307         <Link>Schemas\Loader\Stored Procedures\GetInvalidConstraints.proc.sql</Link>
308     </Build>
309     <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Tables\Constraints.table.sql">
310         <Link>Schemas\Loader\Tables\Constraints.table.sql</Link>
311     </Build>
312     <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Tables\Keys\Constraints.PK_Constraints.pkey.sql">
313         <Link>Schemas\Loader\Tables\Keys\Constraints.PK_Constraints.pkey.sql</Link>
314     </Build>
315     <Build Include="Schemas\dbo\Tables\QBuilds.table.sql" />
316     <Build Include="Schemas\Loader\Tables\ProcessingStatus_QBuild.table.sql" />
317 </ItemGroup>
318 <ItemGroup>
319     <Build Include="$(SrcRoot)\Common\Database\Schemas\Configuration\Stored Procedures\RebuildIndexes.proc.sql">
320         <Link>Schemas\Configuration\Stored Procedures\RebuildIndexes.proc.sql</Link>
321     </Build>
322     <Build Include="$(SrcRoot)\Common\Database\Schemas\Configuration\Stored Procedures\SetBulkLoadMode.proc.sql">
323         <Link>Schemas\Configuration\Stored Procedures\SetBulkLoadMode.proc.sql</Link>
324     </Build>
325     <Build Include="$(SrcRoot)\Common\Database\Schemas\Maintenance\Tables\Updates.table.sql">
326         <Link>Schemas\Maintenance\Tables\Updates.table.sql</Link>
327     </Build>
328     <Build Include="$(SrcRoot)\Common\Database\Schemas\Maintenance\Tables\Keys\Updates.PK_Updates.pkey.sql">
329         <Link>Schemas\Maintenance\Tables\Keys\Updates.PK_Updates.pkey.sql</Link>
330     </Build>
331 </ItemGroup>
332 <ItemGroup>
333     <None Include="Schema Comparisons\LiveChanges.scmp" />
334     <None Include="Post-Deployment\Data\dbo.DepotsMSG.Data.sql" />
335     <None Include="Post-Deployment\Data\dbo.DepotsWindows.data.sql" />
336     <None Include="Post-Deployment\Data\dbo.DepotsWindowsPhone.data.sql" />
337     <None Include="Post-Deployment\Data\dbo.DepotsWindowsServices.data.sql" />
338     <None Include="Post-Deployment\Data\dbo.DepotsOSD.data.sql" />
339     <None Include="Post-Deployment\Data\dbo.DepotsExchange.data.sql" />
340     <None Include="Post-Deployment\Data\dbo.DepotsLync.data.sql" />
341     <None Include="Post-Deployment\Data\dbo.DepotsEE.data.sql" />
342     <None Include="Post-Deployment\Data\dbo.DepotsOffice.data.sql" />
343     <None Include="Post-Deployment\Data\dbo.DepotsOfficeMac.data.sql" />
344     <None Include="Post-Deployment\Data\dbo.DepotsOfficeMac.data.sql" />

```

- File
- Change
- \$/codemine/dev/src
- CodeMine.sln Edit
- \$/codemine/dev/src/Cmc/Database
- CMC.sqlproj Edit
- \$/codemine/dev/src/Cmc/Database/Post-Deplo
- Loader.Constraints.Data.sql Edit
- \$/codemine/dev/src/Cmc/Database/Schemas/dl
- \$/codemine/dev/src/Cmc/Database/Schemas/Lc
- \$/codemine/dev/src/Cmc/Loader/QBuildLoader
- \$/codemine/dev/src/Cmc/Loader/QBuildLoader
- \$/codemine/dev/src/Cmc/Loader/QBuildLoader
- \$/codemine/dev/src/Cmc/Loader/QBuildLoader
- \$/codemine/dev/src/Cmc/Loader/QBuildLoader
- \$/codemine/dev/src/Cmc/Loader/QBuildLoader
- \$/codemine/dev/src/Commo/LoaderBase
- LoaderBase.csproj Edit
- LoaderController.cs Edit
- \$/codemine/dev/src/Commo/LoaderBase/Que
- QueueManager.cs Edit
- \$/codemine/dev/src/Commo/LoaderBase/Rest
- RestCall.cs Add
- \$/codemine/dev/src/Test/CodeMine.Integration
- CodeMine.IntegrationTests.cs Edit
- \$/codemine/dev/src/Test/CodeMine.Integration
- QBuildLoaderSuite.cs Add
- \$/codemine/dev/src/Test/CodeMine.Integration
- qbuild_all.json Add
- qbuild_finished_1.json Add
- qbuild_finished_2.json Add

CMC.sqlproj

Review Title
Adding support for new types of builds

Work Items/Bugs
e.g. 1001,1002

Review Attachments
Attachments will be listed in description.txt. Browse

Description
New feature enabling saving data from new types of builds

Required Reviewers
Recommended reviewers were added as required reviewers. ⓘ ✕
Michaela Greiler ✕ ; Jack Tilford ✕ ;

Optional Reviewers
Optional reviewers as a semicolon-delimited list. Press **Ctrl+K** to resolve.

Resolve reviewers Publish Cancel Collapse

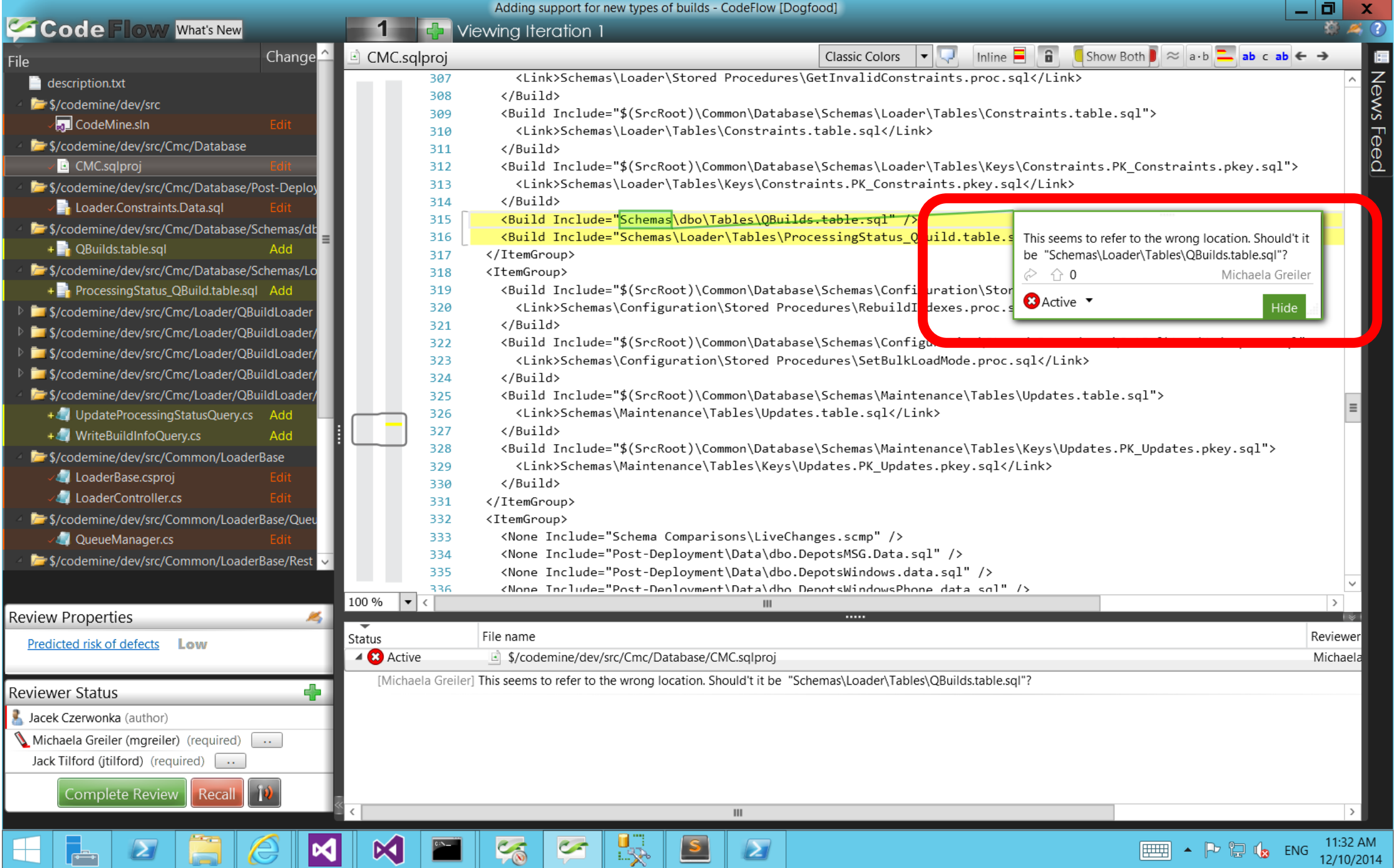
```

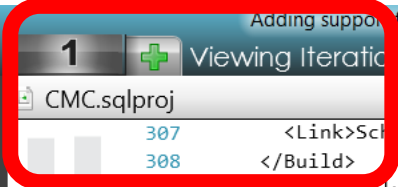
305     </Build>
306     <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Stored Procedures\GetInvalidConstraints.proc.sql">
307         <Link>Schemas\Loader\Stored Procedures\GetInvalidConstraints.proc.sql</Link>
308     </Build>
309     <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Tables\Constraints.table.sql">
310         <Link>Schemas\Loader\Tables\Constraints.table.sql</Link>
311     </Build>
312     <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Tables\Keys\Constraints.PK_Constraints.pkey.sql">
313         <Link>Schemas\Loader\Tables\Keys\Constraints.PK_Constraints.pkey.sql</Link>
314     </Build>
315     <Build Include="Schemas\dbo\Tables\QBuilds.table.sql" />
316     <Build Include="Schemas\Loader\Tables\ProcessingStatus_QBuild.table.sql" />
317 </ItemGroup>
318 <ItemGroup>
319     <Build Include="$(SrcRoot)\Common\Database\Schemas\Configuration\Stored Procedures\RebuildIndexes.proc.sql">
320         <Link>Schemas\Configuration\Stored Procedures\RebuildIndexes.proc.sql</Link>
321     </Build>
322     <Build Include="$(SrcRoot)\Common\Database\Schemas\Configuration\Stored Procedures\SetBulkLoadMode.proc.sql">
323         <Link>Schemas\Configuration\Stored Procedures\SetBulkLoadMode.proc.sql</Link>
324     </Build>
325     <Build Include="$(SrcRoot)\Common\Database\Schemas\Maintenance\Tables\Updates.table.sql">
326         <Link>Schemas\Maintenance\Tables\Updates.table.sql</Link>
327     </Build>
328     <Build Include="$(SrcRoot)\Common\Database\Schemas\Maintenance\Tables\Keys\Updates.PK_Updates.pkey.sql">
329         <Link>Schemas\Maintenance\Tables\Keys\Updates.PK_Updates.pkey.sql</Link>
330     </Build>
331 </ItemGroup>

```

Review Properties

Predicted risk of defects **Low**





File

- description.txt
- \$/codemine/dev/src
- CodeMine.sln Edit
- \$/codemine/dev/src/Cmc/Database
- CMC.sqlproj Edit
- \$/codemine/dev/src/Cmc/Database/Post-Deploy
- Loader.Constraints.Data.sql Edit
- \$/codemine/dev/src/Cmc/Database/Schemas/db
- QBuilds.table.sql Add
- \$/codemine/dev/src/Cmc/Database/Schemas/Lo
- ProcessingStatus_QBuild.table.sql Add
- \$/codemine/dev/src/Cmc/Loader/QBuildLoader
- \$/codemine/dev/src/Cmc/Loader/QBuildLoader/
- \$/codemine/dev/src/Cmc/Loader/QBuildLoader/
- \$/codemine/dev/src/Cmc/Loader/QBuildLoader/
- \$/codemine/dev/src/Cmc/Loader/QBuildLoader/
- UpdateProcessingStatusQuery.cs Add
- WriteBuildInfoQuery.cs Add
- \$/codemine/dev/src/Common/LoaderBase
- LoaderBase.csproj Edit
- LoaderController.cs Edit
- \$/codemine/dev/src/Common/LoaderBase/Queu
- QueueManager.cs Edit
- \$/codemine/dev/src/Common/LoaderBase/Rest

```

307 <Link>Schemas\Loader\Stored Procedures\GetInvalidConstraints.proc.sql</Link>
308 </Build>
309 <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Tables\Constraints.table.sql">
310 <Link>Schemas\Loader\Tables\Constraints.table.sql</Link>
311 </Build>
312 <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Tables\Keys\Constraints.PK_Constraints.pkey.sql">
313 <Link>Schemas\Loader\Tables\Keys\Constraints.PK_Constraints.pkey.sql</Link>
314 </Build>
315 <Build Include="Schemas\dbo\Tables\QBuilds.table.sql" />
316 <Build Include="Schemas\Loader\Tables\ProcessingStatus_QBuild.table.s
317 </ItemGroup>
318 <ItemGroup>
319 <Build Include="$(SrcRoot)\Common\Database\Schemas\Configuration\Stor
320 <Link>Schemas\Configuration\Stored Procedures\RebuildIndices.proc.s
321 </Build>
322 <Build Include="$(SrcRoot)\Common\Database\Schemas\Configuration\Stor
323 <Link>Schemas\Configuration\Stored Procedures\SetBulkLoadMode.proc.
324 </Build>
325 <Build Include="$(SrcRoot)\Common\Database\Schemas\Maintenance\Tables\Updates.table.sql">
326 <Link>Schemas\Maintenance\Tables\Updates.table.sql</Link>
327 </Build>
328 <Build Include="$(SrcRoot)\Common\Database\Schemas\Maintenance\Tables\Keys\Updates.PK_Updates.pkey.sql">
329 <Link>Schemas\Maintenance\Tables\Keys\Updates.PK_Updates.pkey.sql</Link>
330 </Build>
331 </ItemGroup>
332 <ItemGroup>
333 <None Include="Schema Comparisons\LiveChanges.scmp" />
334 <None Include="Post-Deployment\Data\dbo.DepotsMSG.Data.sql" />
335 <None Include="Post-Deployment\Data\dbo.DepotsWindows.data.sql" />
336 <None Include="Post-Deployment\Data\dbo.DepotsWindowsPhone.data.sql" />

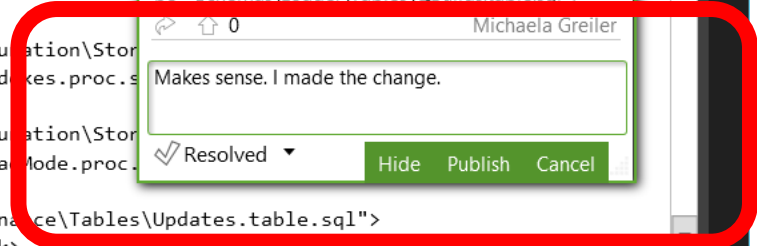
```

This seems to refer to the wrong location. Should't it be "Schemas\Loader\Tables\QBuilds.table.sql"?

0 Michaela Greiler

Makes sense. I made the change.

Resolved Hide Publish Cancel



Review Properties

Predicted risk of defects Low

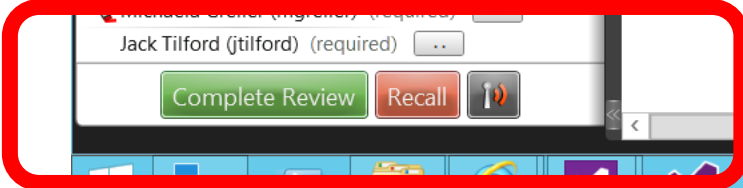
Reviewer Status

Jacek Czerwonka (author)

Michaela Greiler (reviewer) (required)

Jack Tilford (jtilford) (required)

Complete Review Recall



Status	File name	Reviewer
Resolved	\$/codemine/dev/src/Cmc/Database/CMC.sqlproj	Michaela Greiler
[Michaela Greiler] This seems to refer to the wrong location. Should't it be "Schemas\Loader\Tables\QBuilds.table.sql"?		

Changing Task to Definition #2

Edit New issue

Open jacekczzerwonka wants to merge 1 commit into master from refactoring

Conversation 1 Commits 1 Files changed 1 +35 -37

jacekczzerwonka commented 2 minutes ago
Changing "Task" to "Definition"

Changing Task to Definition 19fdb4

jacekczzerwonka commented on the diff 15 seconds ago


```
src/api/pict.cpp View full changes
((14 lines not shown))
21 19 {
22 - Task *task = NULL;
23 - task = new Task();
+ Definition *Definition = NULL;
```

Labels: None yet
Milestone: No milestone
Assignee: No one—assign yourself

Notifications: Unsubscribe

1 participant

```
9      9      void generateModel ( Model *model );
10     10     void deleteModel ( Model *model );
11     11
12     12     -#define NO_CONST_HANDLE(X) const_cast< void * >( X )
13     13     -
14     12     //
15     13     //
16     14     //
17     15
18     16     PICT_HANDLE
19     17     API_SPEC
20     17     -PictCreateTask()
21     18     +PictCreateDefinition()
22     19     {
23     20     - Task *task = NULL;
24     21     - task = new Task();
25     22     + Definition *Definition = NULL;
```

 **jacekczzerwonka** added a note just now Owner ✎ ✕

Hard to read when type name and variable name are the same. Change?




Add a line note

```
26     21     + Definition = new Definition();
27     22
28     23     - return( static_cast<PICT_HANDLE>( task ));
29     24     + return( static_cast<PICT_HANDLE>( Definition ));
30     25     };
31     26     //
32     @@ -33,12 +31,12 @@ void
33     31     API_SPEC
34     32     PictSetRootModel
35     33     (
```

```

  ⚙ ((14 lines not shown))
  21 19 {
  22 - Task *task = NULL;
  23 - task = new Task();
      20 + Definition *Definition = NULL;

```

 **jacekczerwonka** added a note 15 seconds ago Owner  

Hard to read when type name and variable name are the same. Change?

Add a line note

Notifications

 **Unsubscribe**

You're receiving notifications because you authored the thread.

1 participant



 Lock pull request

Add more commits by pushing to the **refactoring** branch on **jacekczerwonka/pict**.



This pull request can be automatically merged.
You can also merge branches on the [command line](#).



Merge pull request



Write Preview

 Markdown supported  Edit in fullscreen

Leave a comment

Attach images by dragging & dropping or [selecting them](#).

Close pull request

Comment

The goals of code reviewing...



Why Code Review?

Find defects

Improve maintainability

Share knowledge

Broadcast progress

❗ It is all about the conversation

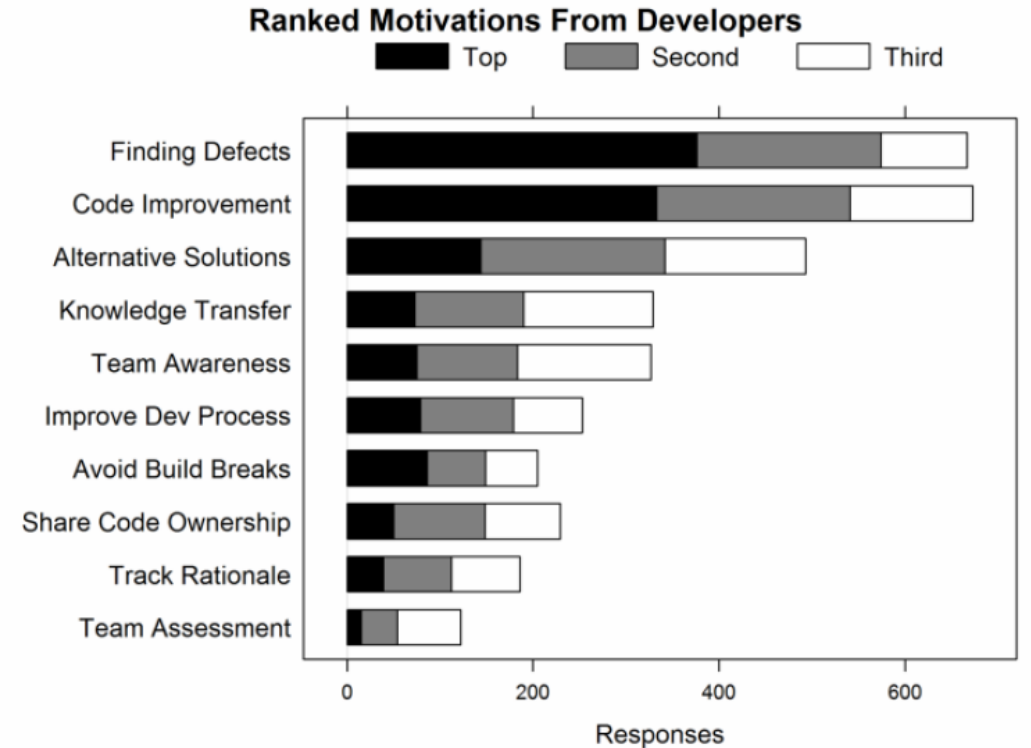
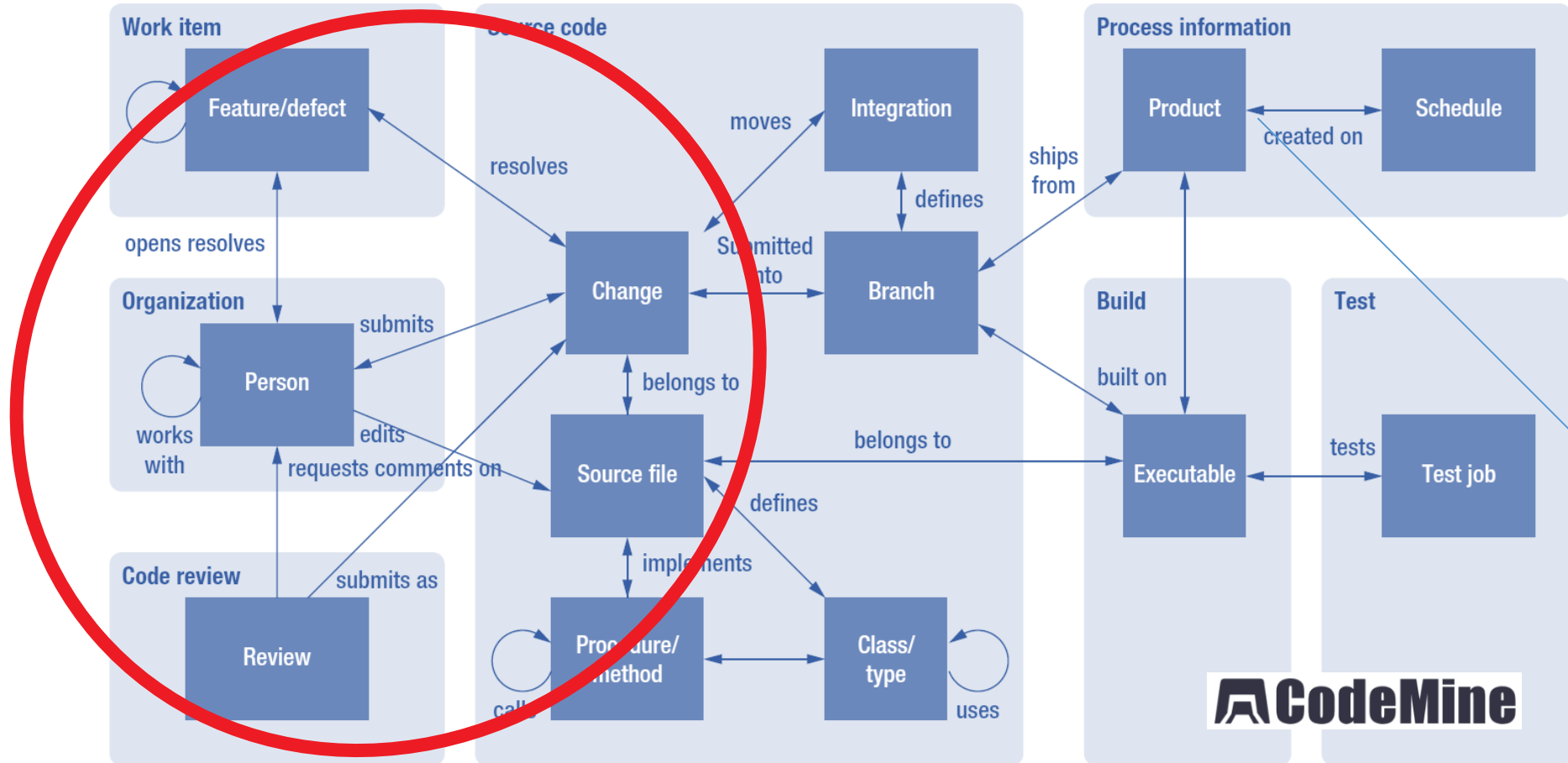


Figure 3. Developers' motivations for code review.

What do we achieve
in practice?









Engineering Data Collection



CodeMine

Study: Comments Classification

Category	Types of issues included	Frequency
Documentation	Comments, naming, style	22.3%
Organization of code	Modularity, location of artifacts, new class, duplicate code, size of methods	15.9%
Solution approach	Alternate algorithm or data structure	8.5%
Validation	Lack of or improper validation	 7.2%
Visual representation	Beautification, indentation, blank links	6.4%
False positive	Not an real issue	4.6%
Defect	Incorrect implemenation or missing functionality	 2.6%
Logical	Control flow or logic issues	 2.3%
Support	Configuration support systems or libraries	2.0%
Interface	Interactions with other components	 1.5%
Resources	Resource initialization, manipulation, and release	 1.3%
Timing	Thread synchronization, races	 0.3%
Other		25.1%

~50% of all

15% of all

Adapted from : M. Mäntylä and C. Lassenius. *What Types of Defects Are Really Discovered in Code Reviews?* IEEE Transactions Software Engineering, 35(3):430–448, 2009

By: Amiangshu Bosu (U of Alabama), Michaela Greiler (TSE), Christian Bird (Microsoft Research Redmond), *Characteristics of Useful Code Reviews: An Empirical Study at Microsoft* (MSR 2015)

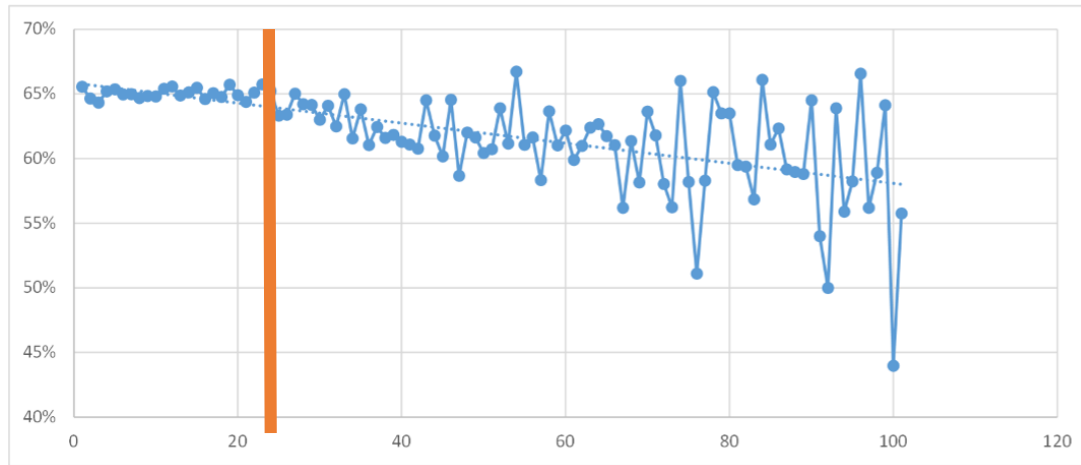
Study: Code Review Usefulness

Category	Types of issues included	Frequency
Documentation	Comments, naming, style	22.3%
Organization of code	Modularity, location of artifacts, new class, duplicate code, size of methods	15.9%
Solution approach	Alternate algorithm or data structure	8.5%
Validation	Lack of or improper validation	7.2%
Visual representation	Beautification, indentation, blank lines	6.4%
False positive	Not a real issue	4.6%
Defect	Incorrect implementation or missing functionality	2.6%
Logical	Control flow or logic issues	2.3%
Support	Configuration support systems or libraries	2.0%
Interface	Interactions with other components	1.5%
Resources	Resource initialization, manipulation, and release	1.3%
Timing	Thread synchronization, races	0.3%
Other		25.1%

Adapted from : *M. Mäntylä and C. Lassenius. What Types of Defects Are Really Discovered in Code Reviews? IEEE Transactions Software Engineering, 35(3):430–448, 2009*

By: Amiangshu Bosu (U of Alabama), Michaela Greiler (TSE), Christian Bird (Microsoft Research Redmond), *Characteristics of Useful Code Reviews: An Empirical Study at Microsoft (MSR 2015)*

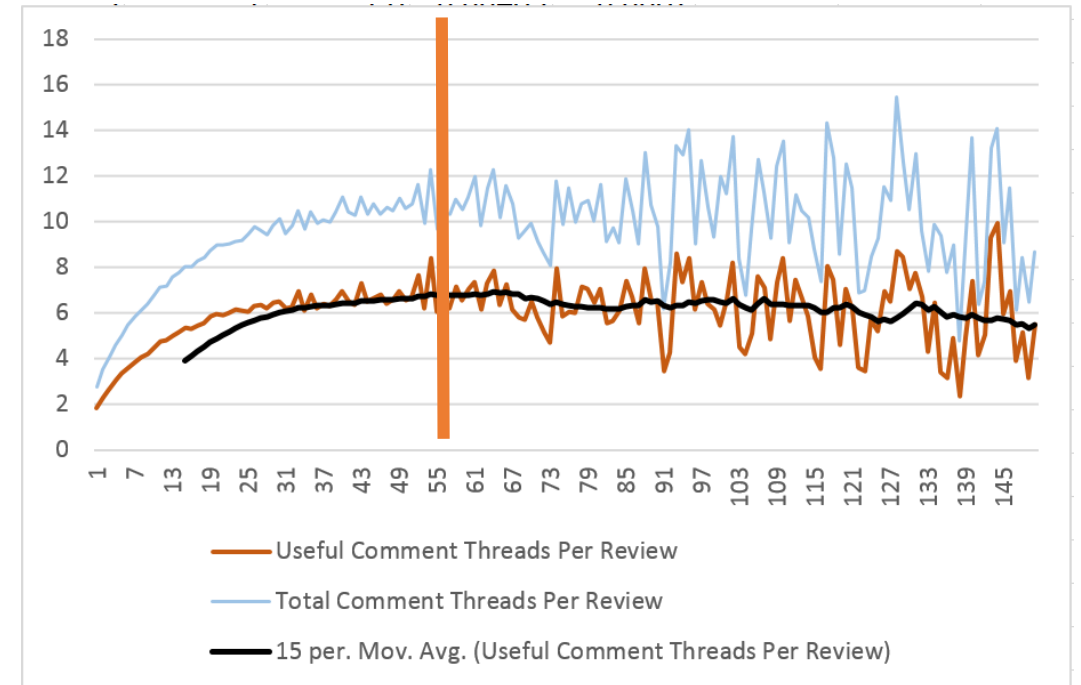
Smaller Reviews Are Better



Comment usefulness ratio vs. number of files in a reviewed change

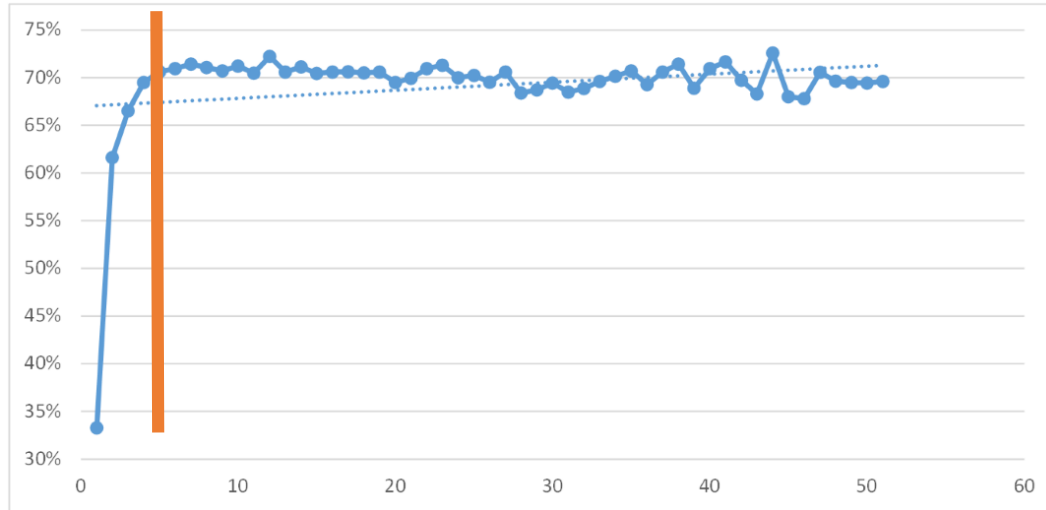
Anecdotaly: smaller reviews are “better”

From data: ≤ 20 files implies usefulness stability and predictability



Absolute number of useful comments grows with size of review until 25-30 files, steady until 55-65 and then starts going down

Relevant Experience Makes for Better Reviewers

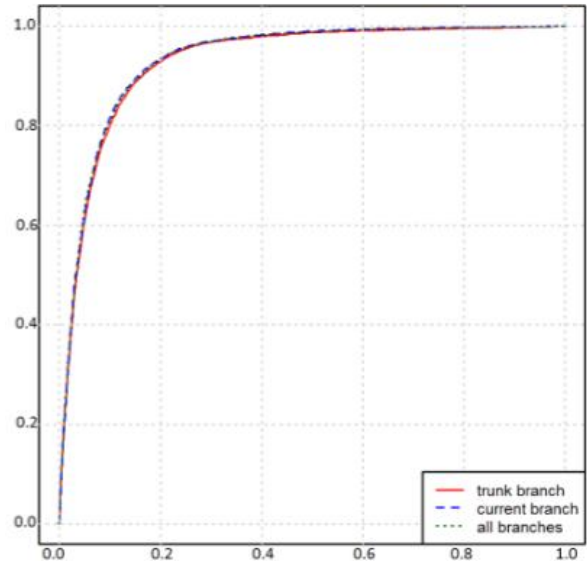


Reviewer's comment usefulness vs. number of previous reviews on a changed file

Reviewers with prior experience with the changed file produce much more useful feedback

New reviewers learn fast but need 6-12 months to be as productive as the rest of the team

Risk of Defects In a Change Can Be Predicted



Vista SP1 QFE View Details and Notes 1

Change Summary

Fix Regression Risk: **Very High** (Probability of regression > 50%)

Binary	Arch Layer	Total/Post-LKG ch
p.dll	44 (range:0-64)	2/2
Source File		
▶ base\diagnosis\pdata.c		
▼ base\diagnosis\putil.c		
Function		Complexity
Connect		30->32
▶ base\diagnosis\query.c		
s.ocx	45 (range:0-64)	1/1
r.dll	11 (range:0-64)	2/2

Prior success with large-scale defect prediction

Expose risk prediction in code review to change the reviewer behavior

Predicting Risk of Pre-Release Code Changes with CheckinMentor, A. Tarvo, N. Nagappan, T. Zimmermann, T. Bhat, J. Czerwonka

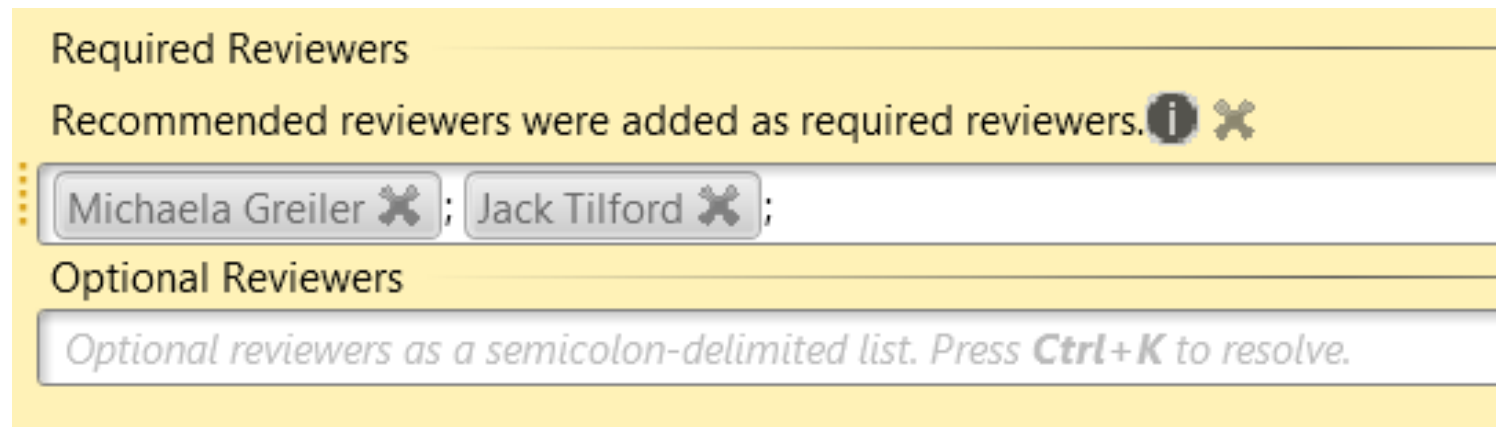
CRANE: Failure Prediction, Change Analysis and Test Prioritization in Practice - Experiences from Windows, J. Czerwonka, R. Das, N. Nagappan, A. Tarvo, A. Teterev

Improving the tools

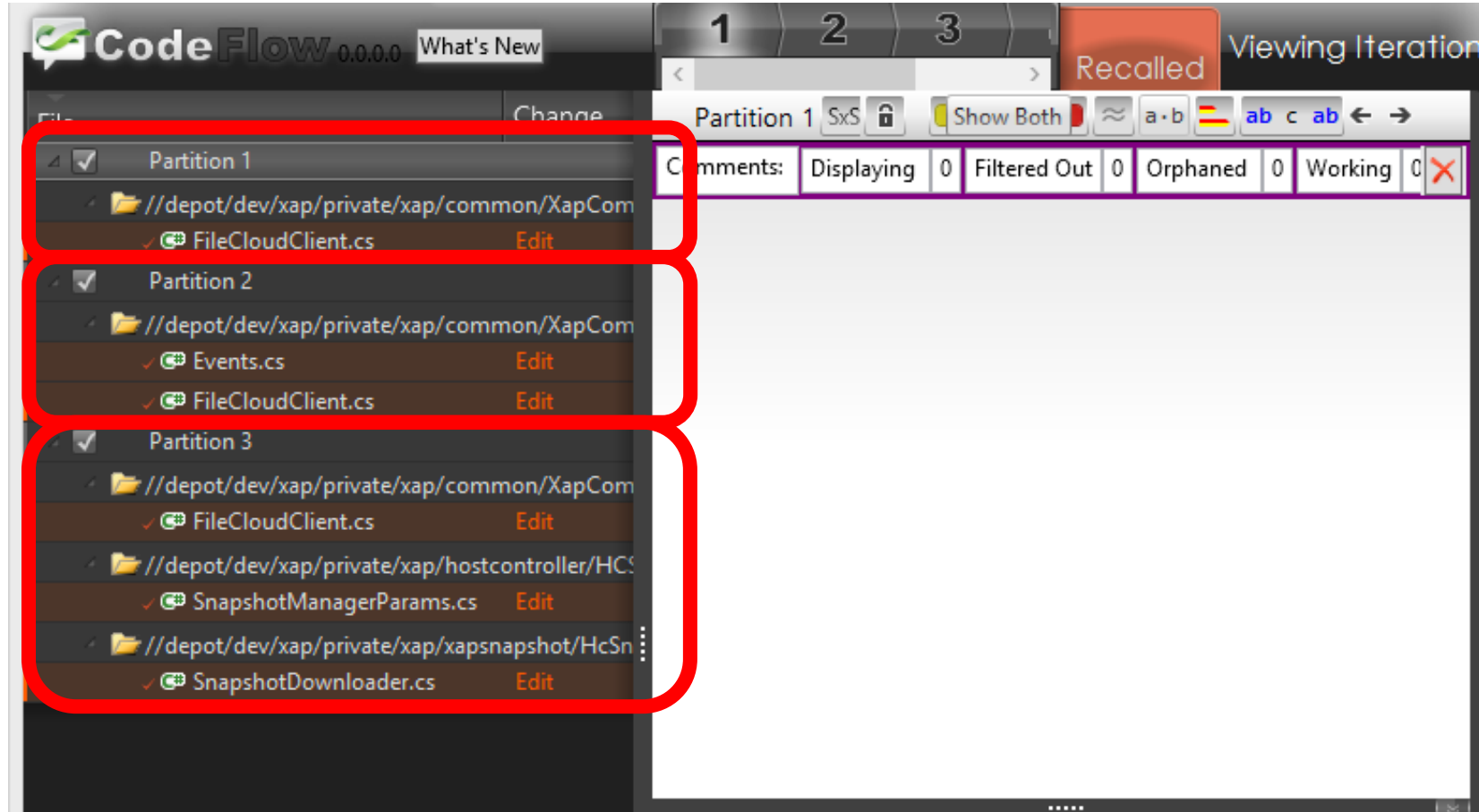


Feature #1: Reviewer Recommendations

- Find potential reviewers based on their previous history with the code
- Consider number of changes and time since last activity
- Default is two reviewers based on most common practice and usefulness data

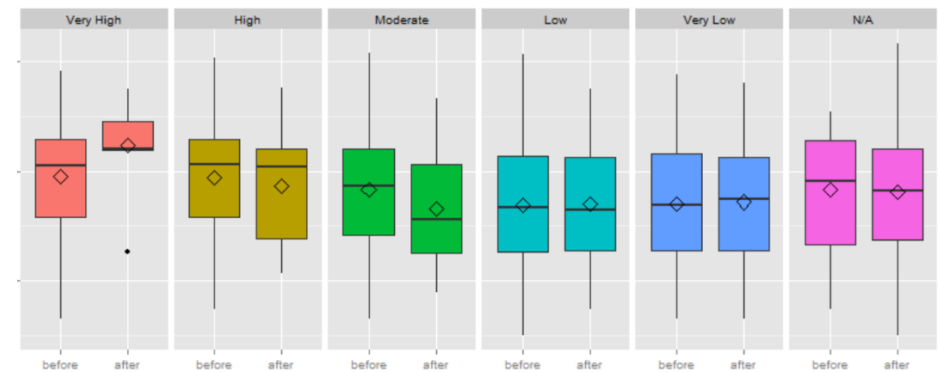
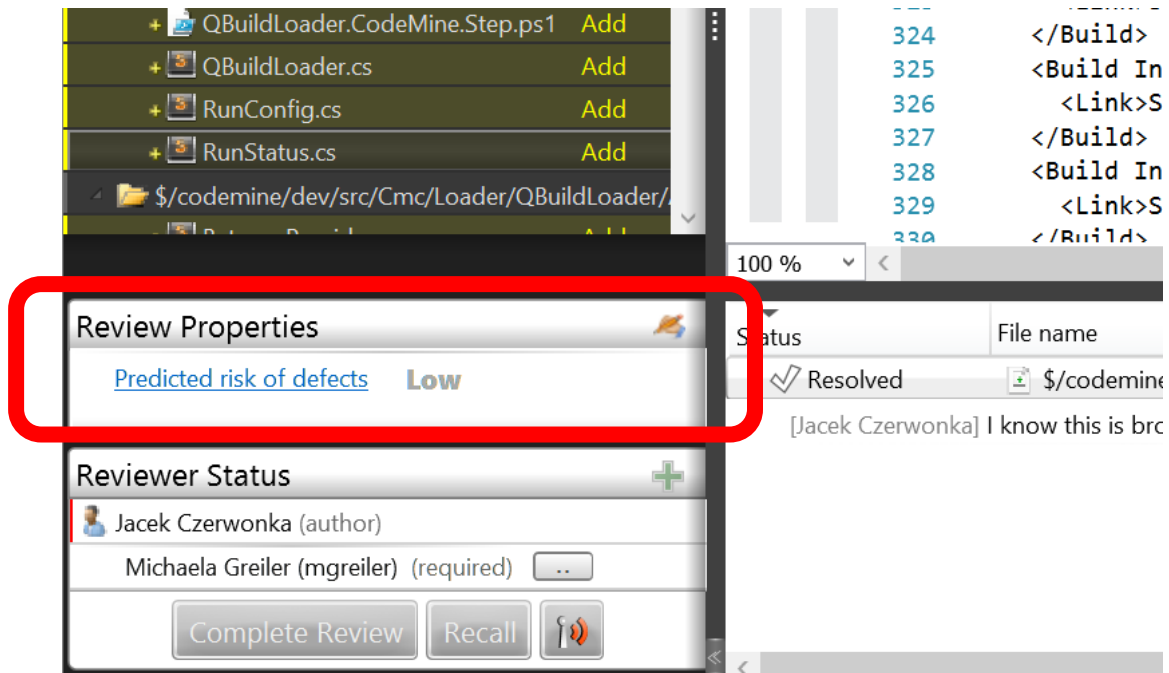


Feature #2: Change Decomposition



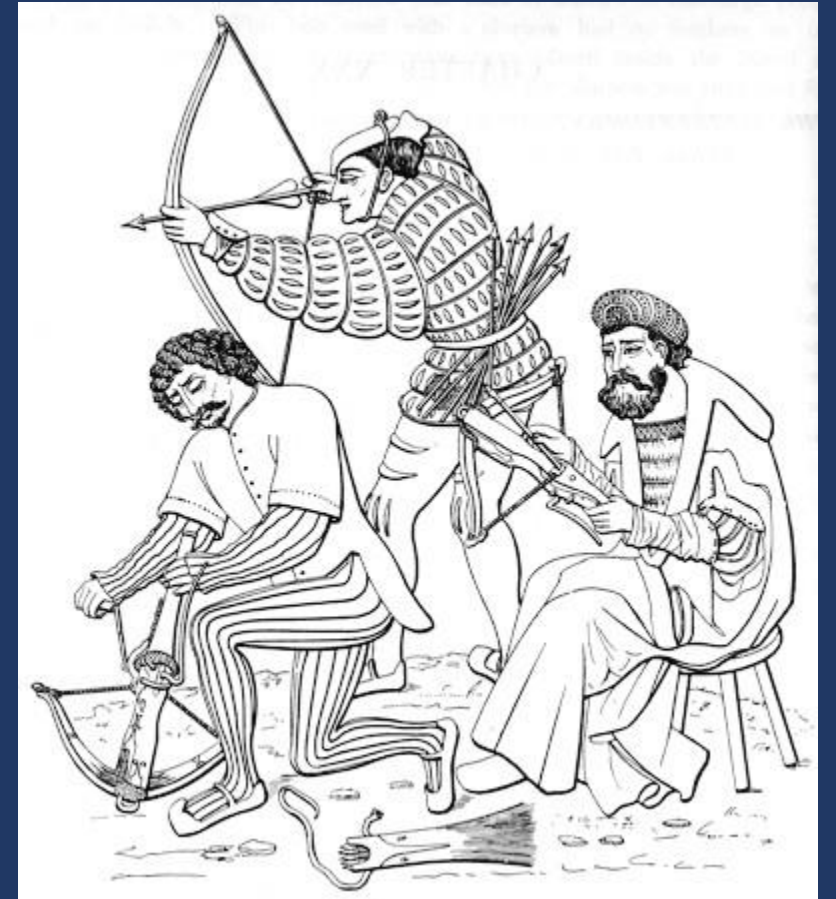
By: Shuvendu Lahiri, Mike Barnett, Christian Bird, Jack Tilford (Microsoft Research Redmond and TSE)

Feature #3: Change Risk Prediction

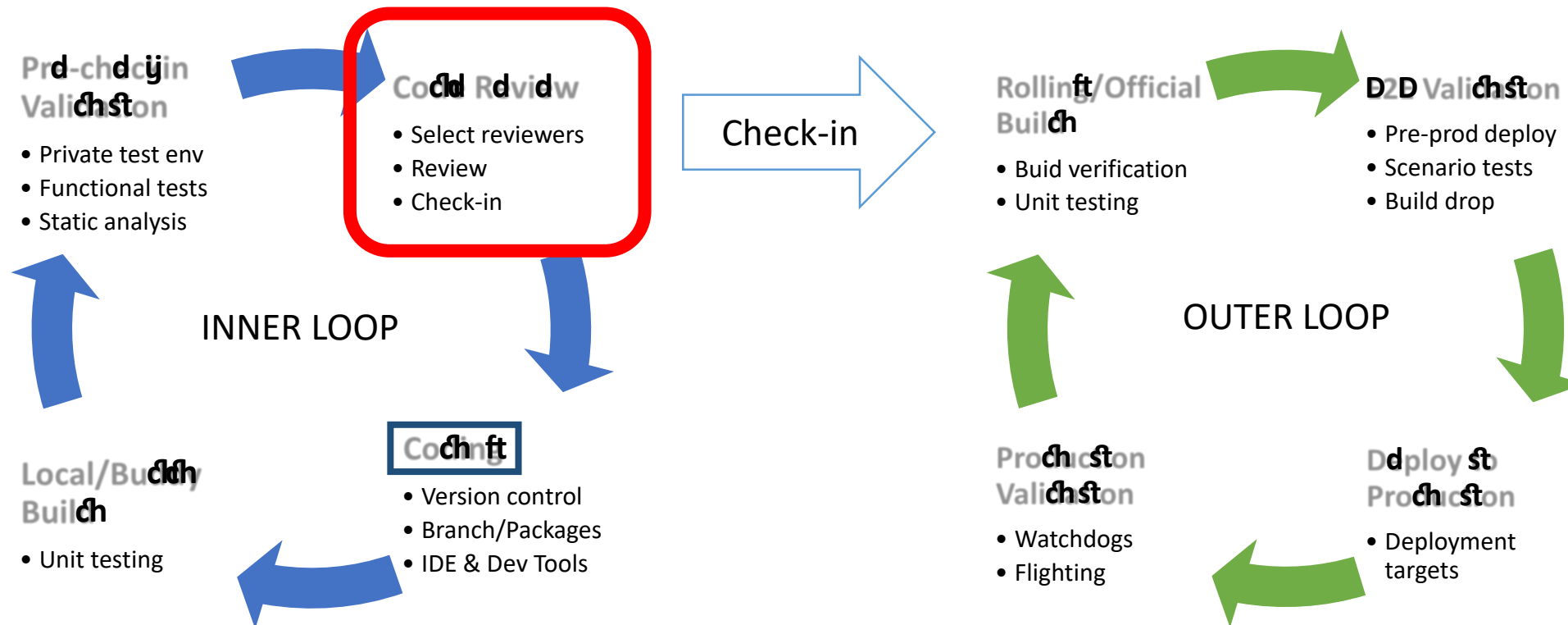


Time to first sign-off before and after enabling "change risk" feature







Improving the workflow



Code Reviews in Engineering Workflow



Study: Comments Classification

Category	Types of issues included	Frequency
Documentation	Comments, naming, style	22.3%
Organization of code	Modularity, location of artifacts, new class, duplicate code, size of methods	15.9%
Solution approach	Alternate algorithm or data structure	8.5%
Validation	Lack of or improper validation	 7.2%
Visual representation	Beautification, indentation, blank links	6.4%
False positive	Not an real issue	4.6%
Defect	Incorrect implemenation or missing functionality	 2.6%
Logical	Control flow or logic issues	 2.3%
Support	Configuration support systems or libraries	2.0%
Interface	Interactions with other components	 1.5%
Resources	Resource initialization, manipulation, and release	 1.3%
Timing	Thread synchronization, races	 0.3%
Other		25.1%

~50% of all

15% of all

Adapted from : M. Mäntylä and C. Lassenius. *What Types of Defects Are Really Discovered in Code Reviews?* IEEE Transactions Software Engineering, 35(3):430–448, 2009

By: Amiangshu Bosu (U of Alabama), Michaela Greiler (TSE), Christian Bird (Microsoft Research Redmond), *Characteristics of Useful Code Reviews: An Empirical Study at Microsoft* (MSR 2015)

Code Reviewing: It Takes Time and Effort

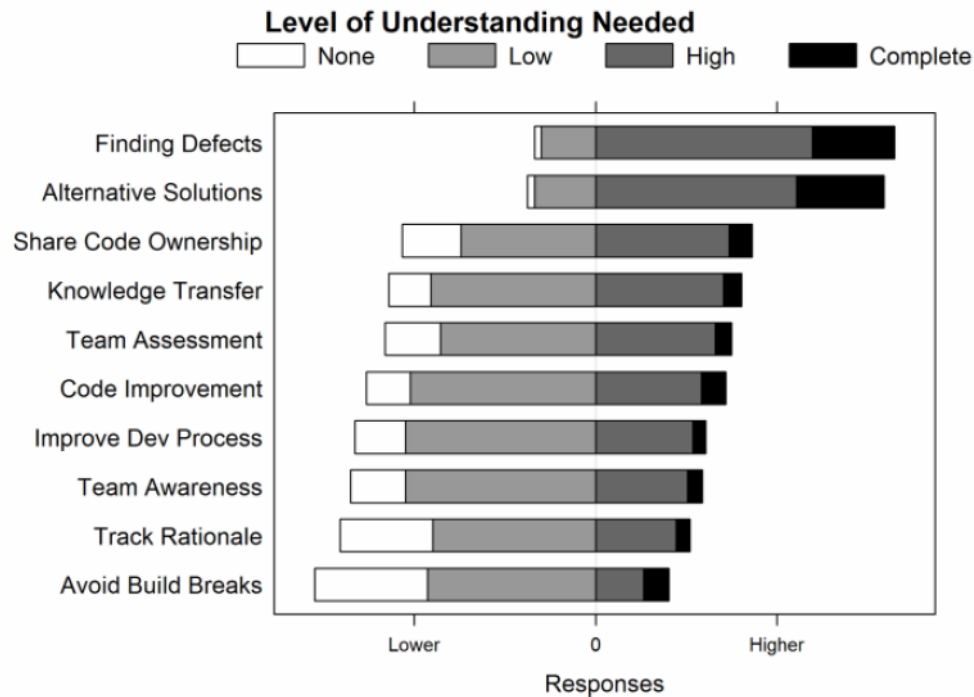


Figure 5. Developers' responses in surveys of the amount of code understanding for code review outcomes.

Alberto Bacchelli, Christian Bird. *Expectations, Outcomes, and Challenges Of Modern Code Review*

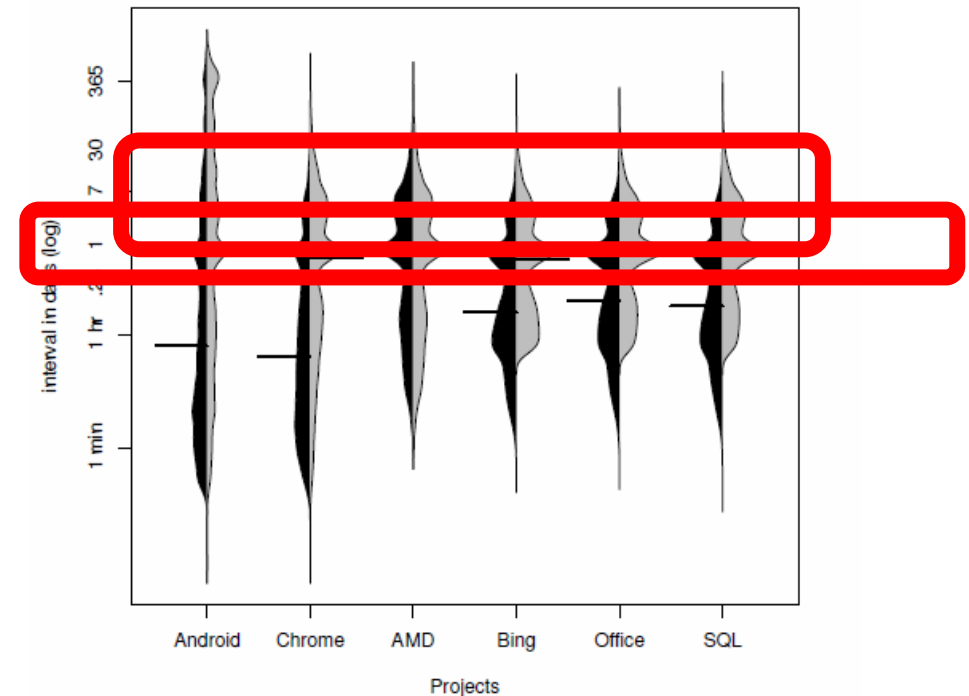


Figure 1: First Response on left (we do not have first response data for AMD) and Full interval on right

Peter C. Rigby, and Christian Bird. *Convergent contemporary software peer review practices*. In Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering, 2013, ESEC/FSE 2013, ACM, pp. 202–212

The Importance of Speed

General

ddch ij Please implement a "virtual whip" feature to ping people who are dragging their feet on a review.

d d xyz-a723ea9fdc14440586656f9

d st test

R d st 123456

Effects of delaying:

- Process stalls; impact on dependents
- Costly context switches for engineers

Code Reviewing is a Social Process

- Waiting is not just due to lack of time:
 - I'm expected to participate but I'm not quite sure how. I'll wait until someone else starts.*
 - There are a lot of outstanding comments already. I'll wait until the next version.*
- Code reviewing in social context
 - Reviewing can be uncomfortable for authors and reviewers
 - Team's hierarchy influence the outcome
- How many engineers does it take to do a proper code review?
 - Sometimes reviewers are added as a courtesy

Code Reviewing is Social

Review completed

Submitted in changelist:3561857

Open in: [\[CodeFlow\]](#) [\[Dogfood\]](#) [\[Browser\]](#) [\[Visual Studio\]](#)

Author: 

Reviewer	Type	Status
Newton S...	Required	SignedOff
Vinod S...	Required	SignedOff
Erik (...	Required	SignedOff
Yancho V...	Required	Started
Hamed B...	Required	NotStarted
CodeFlow Listener (b...	Optional	SignedOff
Q Build Team - Internal (d...	Optional	InviteOnly

Description:

Fix Bug 168324. Zero build outputs get Cached and persist through sessions causing build failures

Code Reviews Are Not Free

Benefits

- More maintainable designs
- More consistent code base
- Knowledge sharing
- Better awareness of changes
- Additional defects found

Costs

- Time spent by reviewers
- Time spent by author addressing feedback
- Time spent by change waiting in process

Costs

Benefits

Opportunities for Optimization

□ **st** **st r jr**

- Syntax, style conformance, static analysis

jd d r d d ch dr dr

- When is it required to have a senior engineer?
- Since we mostly find maintainability issues, who is best at that?
- How to best prioritize work for reviewers?

d cl r r

- Code review after submission not before?

D **st r r**

- Auto-submit after required reviewers signed off?

jd d d ft st

- Does every change need the same level of reviewing?
- Which types of changes do not benefit little from code reviewing?

R ft

st ij

ft dr d d d r st ch ft

ch d

cl d d r cl d

Summary

Code reviews do not find as many bugs as you may think (they are still very useful)

There is a discrepancy between what developers aim for and what the process does

You can't use code reviews as a replacement for other verification techniques

Code reviewing is costly as often the longest and most variable part of code integration

Code reviewing is a skill that needs time to hone

Social aspects of code reviewing make it a complex process

Data can influence changes to tools but verify the outcome

Applying the same process to all commits is wasteful but too early to make changes

To improve further we need to understand the process even more



Tools for Software Engineers

jacekcz@microsoft.com
@jacekcz

Mission: “Enable Microsoft to accelerate its software development”

TSE contributes to and innovates on major parts of the engineering system

TSE runs many of the engineering services and works with all engineering teams

TSE collaborates closely with Microsoft Research teams

Collaboration with academia: visiting researchers and PhD interns (10 this summer)



Wolfram Schulte



Michaela Greiler



Kim Herzig



Suresh
Thummalapenta



Nikolai Tillmann



Trevor Carnahan

<http://research.microsoft.com/tse>