Microsoft

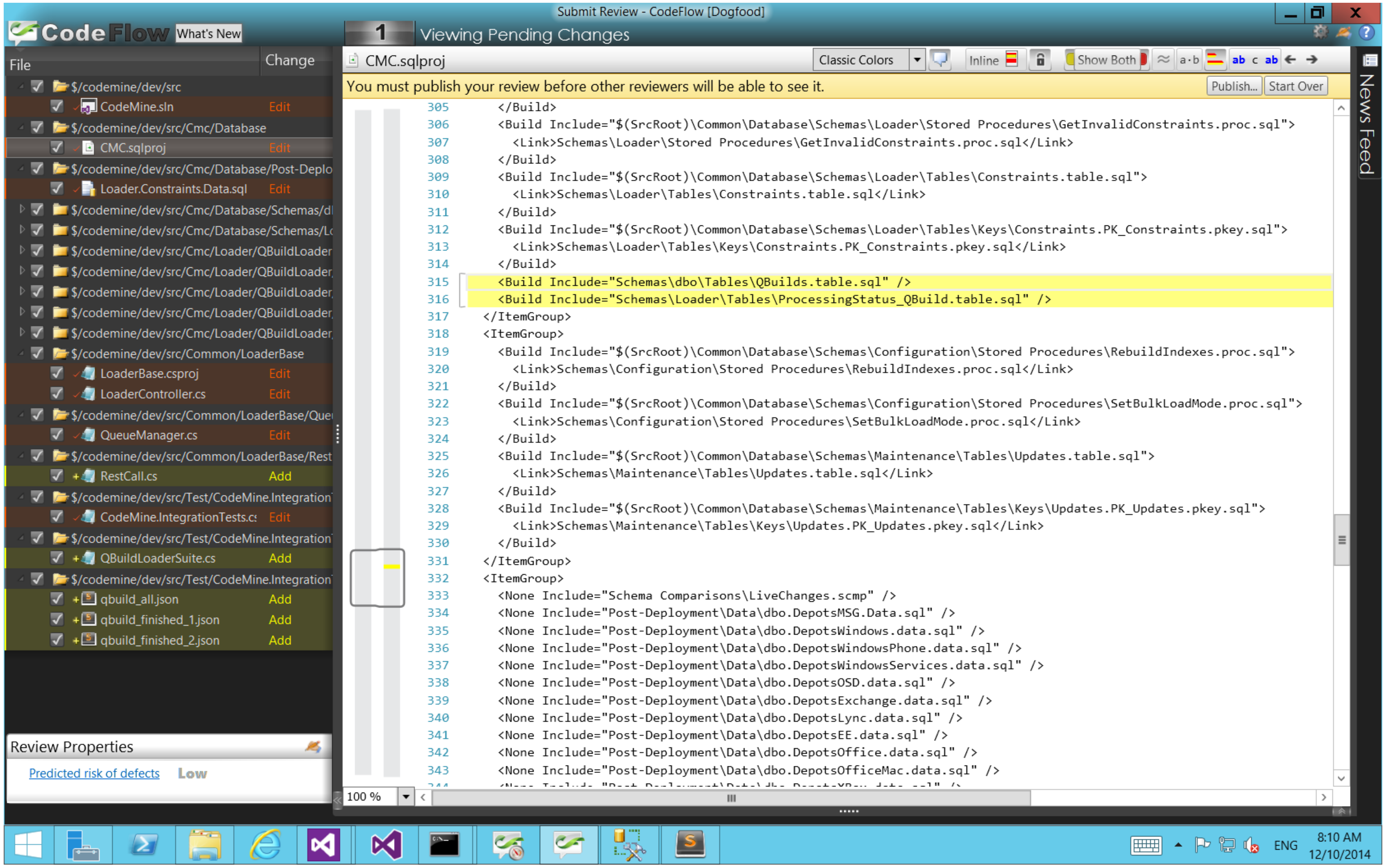
Bridging Gaps between Research and Industry in Software Engineering

Jacek Czerwonka

jacekcz@microsoft.com / jaccz@github.com

Developer Services, Microsoft

http://research.microsoft.com/tse

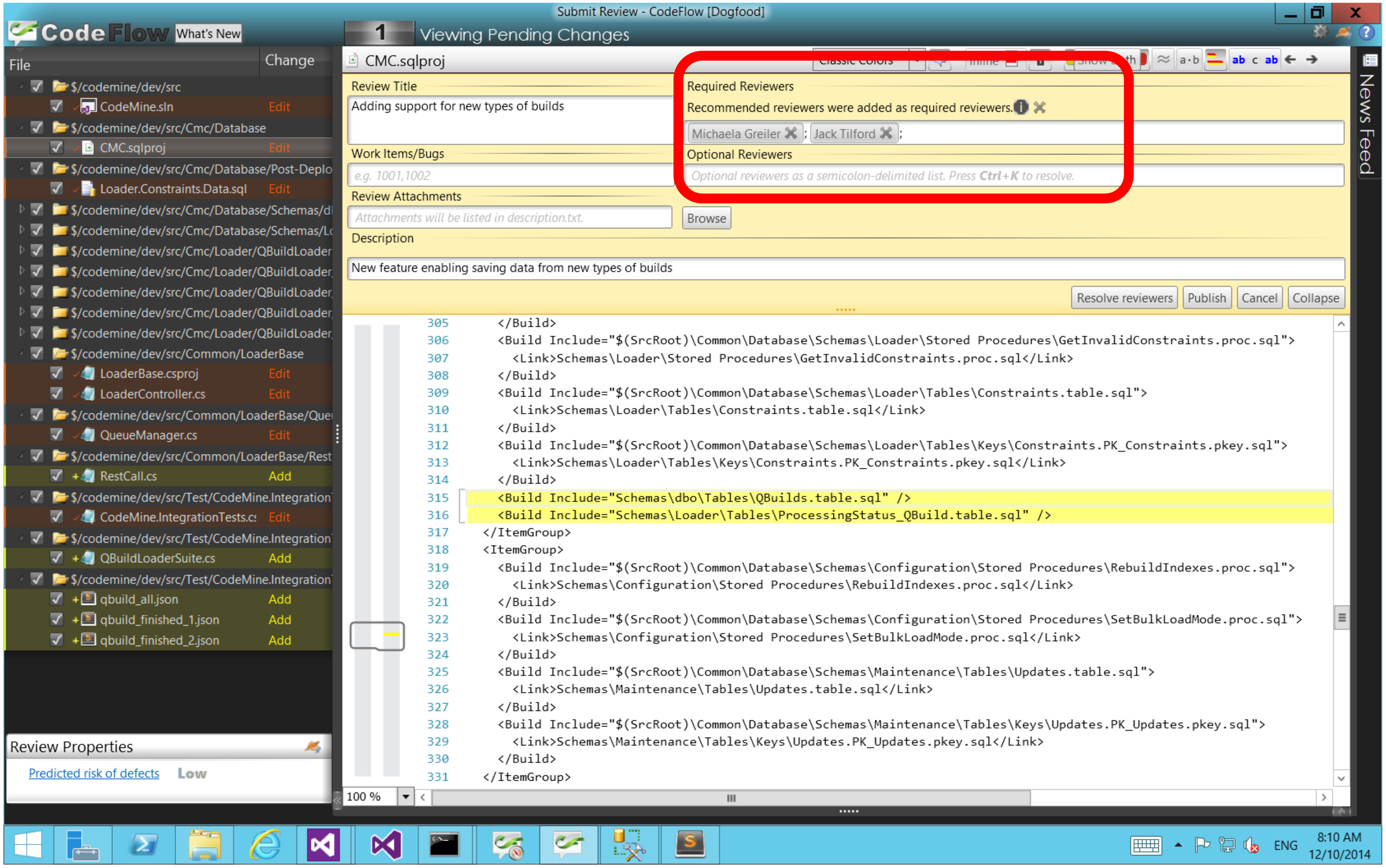


You must publish your review before other reviewers will be able to see it.

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/d	
$/codemine/dev/src/Cmc/Database/Schemas/L	
$/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

```
      </Build>
      <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Stored Procedures\GetInvalidConstraints.proc.sql">
        <Link>Schemas\Loader\Stored Procedures\GetInvalidConstraints.proc.sql</Link>
      </Build>
      <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Tables\Constraints.table.sql">
        <Link>Schemas\Loader\Tables\Constraints.table.sql</Link>
      </Build>
      <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Tables\Keys\Constraints.PK_Constraints.pkey.sql">
        <Link>Schemas\Loader\Tables\Keys\Constraints.PK_Constraints.pkey.sql</Link>
      </Build>
      <Build Include="Schemas\dbo\Tables\QBuilds.table.sql" />
      <Build Include="Schemas\Loader\Tables\ProcessingStatus_QBuild.table.sql" />
    </ItemGroup>
    <ItemGroup>
      <Build Include="$(SrcRoot)\Common\Database\Schemas\Configuration\Stored Procedures\RebuildIndexes.proc.sql">
        <Link>Schemas\Configuration\Stored Procedures\RebuildIndexes.proc.sql</Link>
      </Build>
      <Build Include="$(SrcRoot)\Common\Database\Schemas\Configuration\Stored Procedures\SetBulkLoadMode.proc.sql">
        <Link>Schemas\Configuration\Stored Procedures\SetBulkLoadMode.proc.sql</Link>
      </Build>
      <Build Include="$(SrcRoot)\Common\Database\Schemas\Maintenance\Tables\Updates.table.sql">
        <Link>Schemas\Maintenance\Tables\Updates.table.sql</Link>
      </Build>
      <Build Include="$(SrcRoot)\Common\Database\Schemas\Maintenance\Tables\Keys\Updates.PK_Updates.pkey.sql">
        <Link>Schemas\Maintenance\Tables\Keys\Updates.PK_Updates.pkey.sql</Link>
      </Build>
    </ItemGroup>
    <ItemGroup>
      <None Include="Schema Comparisons\LiveChanges.scmp" />
      <None Include="Post-Deployment\Data\dbo.DepotsMSG.Data.sql" />
      <None Include="Post-Deployment\Data\dbo.DepotsWindows.data.sql" />
      <None Include="Post-Deployment\Data\dbo.DepotsWindowsPhone.data.sql" />
      <None Include="Post-Deployment\Data\dbo.DepotsWindowsServices.data.sql" />
      <None Include="Post-Deployment\Data\dbo.DepotsOSD.data.sql" />
      <None Include="Post-Deployment\Data\dbo.DepotsExchange.data.sql" />
      <None Include="Post-Deployment\Data\dbo.DepotsLync.data.sql" />
      <None Include="Post-Deployment\Data\dbo.DepotsEE.data.sql" />
      <None Include="Post-Deployment\Data\dbo.DepotsOffice.data.sql" />
      <None Include="Post-Deployment\Data\dbo.DepotsOfficeMac.data.sql" />
```

Predicted risk of defects   Low



$/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/d
$/codemine/dev/src/Cmc/Database/Schemas/L
$/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.c 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

Review Title
Adding support for new types of builds

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

Work Items/Bugs
e.g. 1001,1002

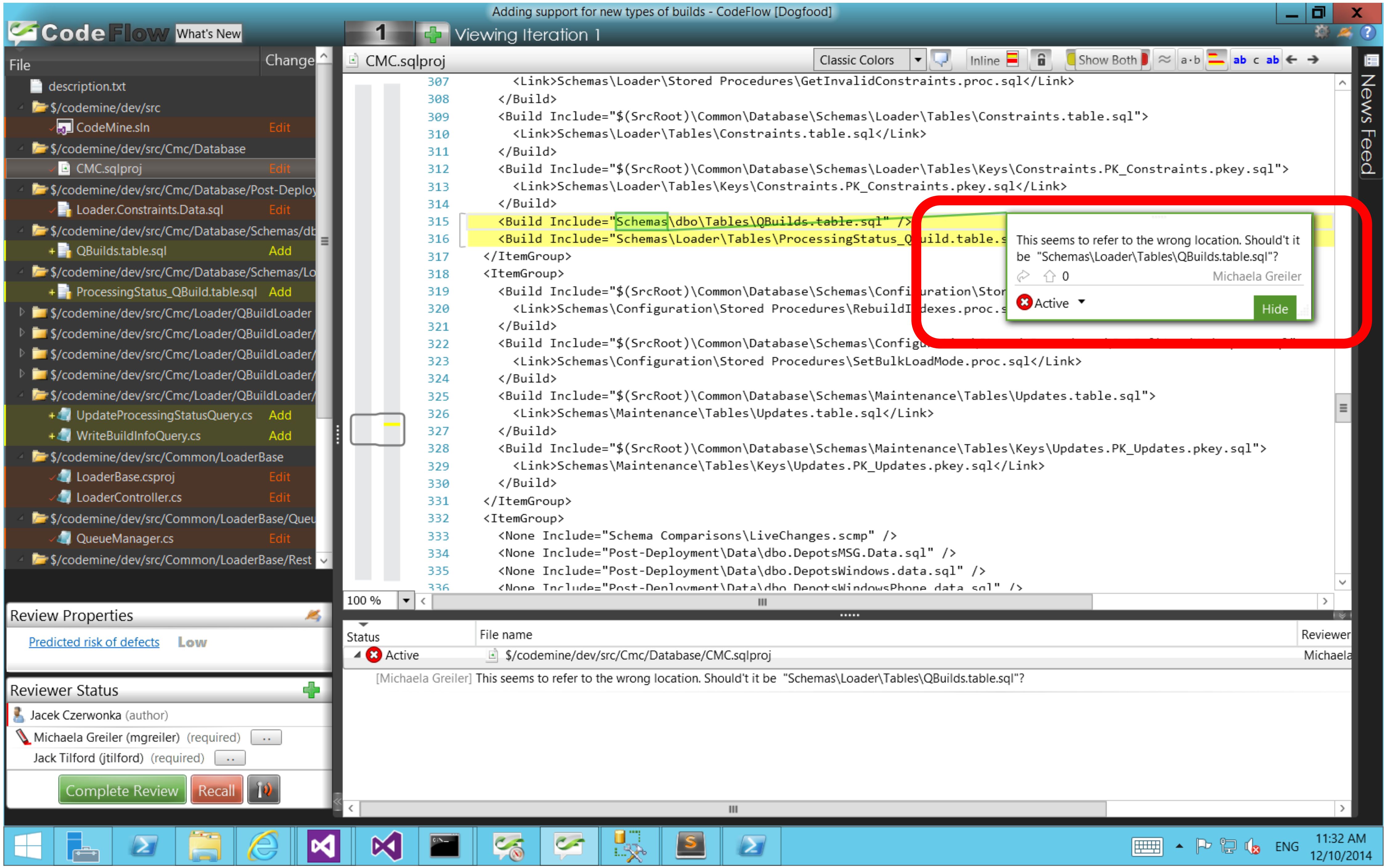
Optional Reviewers
Optional reviewers as a semicolon-delimited list. Press Ctrl+K to resolve.

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

Description
New feature enabling saving data from new types of builds

Resolve reviewers | Publish | Cancel | Collapse

```
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Stored Procedures\GetInvalidConstraints.proc.sql">
      <Link>Schemas\Loader\Stored Procedures\GetInvalidConstraints.proc.sql</Link>
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Tables\Constraints.table.sql">
      <Link>Schemas\Loader\Tables\Constraints.table.sql</Link>
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Tables\Keys\Constraints.PK_Constraints.pkey.sql">
      <Link>Schemas\Loader\Tables\Keys\Constraints.PK_Constraints.pkey.sql</Link>
    </Build>
    <Build Include="Schemas\dbo\Tables\QBuilds.table.sql" />
    <Build Include="Schemas\Loader\Tables\ProcessingStatus_QBuild.table.sql" />
  </ItemGroup>
  <ItemGroup>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Configuration\Stored Procedures\RebuildIndexes.proc.sql">
      <Link>Schemas\Configuration\Stored Procedures\RebuildIndexes.proc.sql</Link>
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Configuration\Stored Procedures\SetBulkLoadMode.proc.sql">
      <Link>Schemas\Configuration\Stored Procedures\SetBulkLoadMode.proc.sql</Link>
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Maintenance\Tables\Updates.table.sql">
      <Link>Schemas\Maintenance\Tables\Updates.table.sql</Link>
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Maintenance\Tables\Keys\Updates.PK_Updates.pkey.sql">
      <Link>Schemas\Maintenance\Tables\Keys\Updates.PK_Updates.pkey.sql</Link>
    </Build>
  </ItemGroup>
```


Adding support for new types of builds - CodeFlow [Dogfood]

CodeFlow What's New

1 Viewing Iteration 1

File	Change
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	

CMC.sqlproj — Classic Colors — Inline — Show Both

```
      <Link>Schemas\Loader\Stored Procedures\GetInvalidConstraints.proc.sql</Link>
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Tables\Constraints.table.sql">
      <Link>Schemas\Loader\Tables\Constraints.table.sql</Link>
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Tables\Keys\Constraints.PK_Constraints.pkey.sql">
      <Link>Schemas\Loader\Tables\Keys\Constraints.PK_Constraints.pkey.sql</Link>
    </Build>
    <Build Include="Schemas\dbo\Tables\QBuilds.table.sql" />
    <Build Include="Schemas\Loader\Tables\ProcessingStatus_QBuild.table.s
  </ItemGroup>
  <ItemGroup>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Configuration\Stor
      <Link>Schemas\Configuration\Stored Procedures\RebuildIndexes.proc.s
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Config
      <Link>Schemas\Configuration\Stored Procedures\SetBulkLoadMode.proc.sql</Link>
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Maintenance\Tables\Updates.table.sql">
      <Link>Schemas\Maintenance\Tables\Updates.table.sql</Link>
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Maintenance\Tables\Keys\Updates.PK_Updates.pkey.sql">
      <Link>Schemas\Maintenance\Tables\Keys\Updates.PK_Updates.pkey.sql</Link>
    </Build>
  </ItemGroup>
  <ItemGroup>
    <None Include="Schema Comparisons\LiveChanges.scmp" />
    <None Include="Post-Deployment\Data\dbo.DepotsMSG.Data.sql" />
    <None Include="Post-Deployment\Data\dbo.DepotsWindows.data.sql" />
    <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
Active Hide

100 %

Status	File name	Reviewer
Active	$/codemine/dev/src/Cmc/Database/CMC.sqlproj	Michaela

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

Review Properties

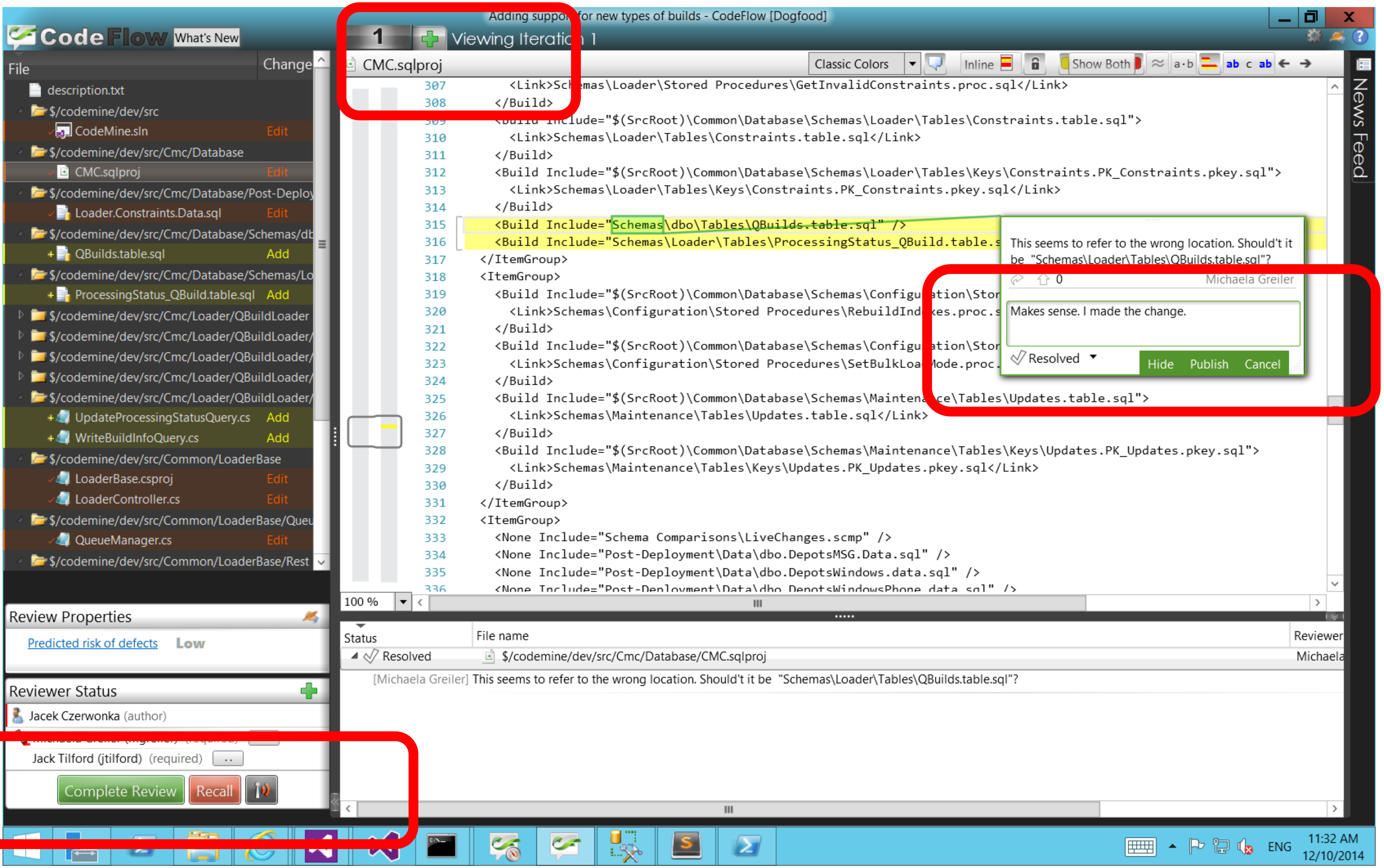
Predicted risk of defects Low

Reviewer Status

Jacek Czerwonka (author)
Michaela Greiler (mgreiler) (required)
Jack Tilford (jtilford) (required)

Complete Review Recall

ENG 11:32 AM 12/10/2014



File	Change
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	

```
      <Link>Schemas\Loader\Stored Procedures\GetInvalidConstraints.proc.sql</Link>
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Tables\Constraints.table.sql">
      <Link>Schemas\Loader\Tables\Constraints.table.sql</Link>
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Loader\Tables\Keys\Constraints.PK_Constraints.pkey.sql">
      <Link>Schemas\Loader\Tables\Keys\Constraints.PK_Constraints.pkey.sql</Link>
    </Build>
    <Build Include="Schemas\dbo\Tables\QBuilds.table.sql" />
    <Build Include="Schemas\Loader\Tables\ProcessingStatus_QBuild.table.s
  </ItemGroup>
  <ItemGroup>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Configuration\Stor
      <Link>Schemas\Configuration\Stored Procedures\RebuildIndexes.proc.s
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Configuration\Stor
      <Link>Schemas\Configuration\Stored Procedures\SetBulkLoadMode.proc.
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Maintenance\Tables\Updates.table.sql">
      <Link>Schemas\Maintenance\Tables\Updates.table.sql</Link>
    </Build>
    <Build Include="$(SrcRoot)\Common\Database\Schemas\Maintenance\Tables\Keys\Updates.PK_Updates.pkey.sql">
      <Link>Schemas\Maintenance\Tables\Keys\Updates.PK_Updates.pkey.sql</Link>
    </Build>
  </ItemGroup>
  <ItemGroup>
    <None Include="Schema Comparisons\LiveChanges.scmp" />
    <None Include="Post-Deployment\Data\dbo.DepotsMSG.Data.sql" />
    <None Include="Post-Deployment\Data\dbo.DepotsWindows.data.sql" />
    <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

Status	File name	Reviewer
Resolved	$/codemine/dev/src/Cmc/Database/CMC.sqlproj	Michaela

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

Review Properties

Predicted risk of defects Low

Reviewer Status

Jacek Czerwonka (author)
Michaela Greiler (mgreiler) (required)
Jack Tilford (jtilford) (required)

Code Review Research

Why Code Review?

Find defects

Improve maintainability

Share knowledge

Broadcast progress

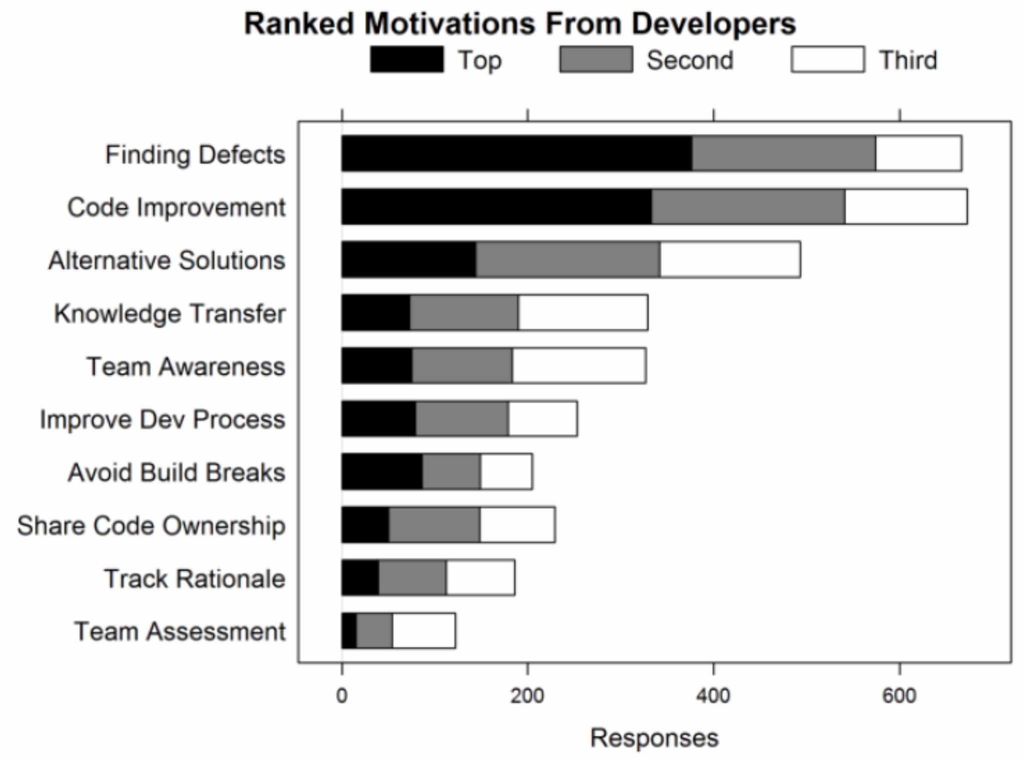


Figure 3. Developers' motivations for code review.

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

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 linkes	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)

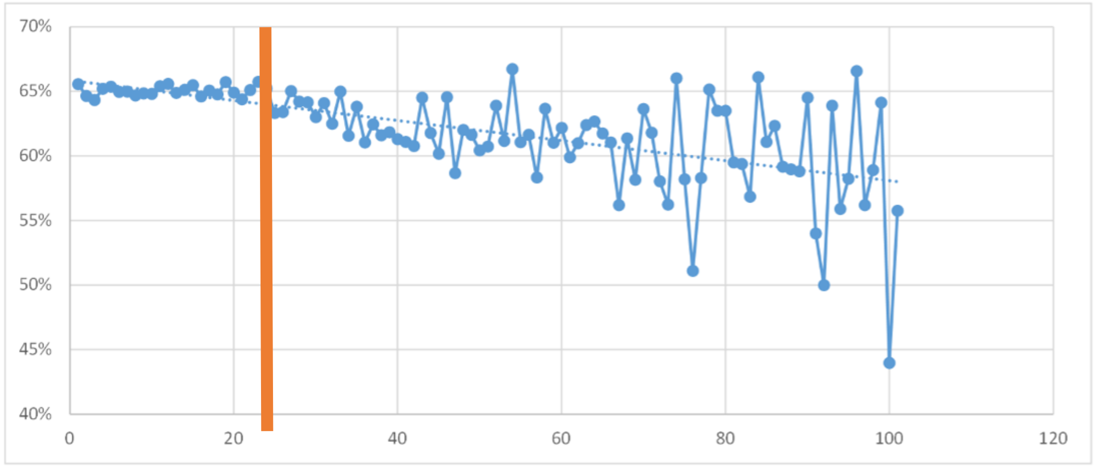
Study: Code Review Usefulness

Category	Types of issues included	Frequency	% Useful
Documentation	Comments, naming, style	22.3%	77.0%
Organization of code	Modularity, location of artifacts, new class, duplicate code, size of methods	15.9%	87.1%
Solution approach	Alternate algorithm or data structure	8.5%	72.3%
Validation	Lack of or improper validation	7.2%	92.9%
Visual representation	Beautification, indentation, blank linkes	6.4%	68.0%
False positive	Not an real issue	4.6%	0.0%
Defect	Incorrect implemenation or missing functionality	2.6%	90.0%
Logical	Control flow or logic issues	2.3%	100.0%
Support	Configuration support systems or libraries	2.0%	87.5%
Interface	Interactions with other components	1.5%	100.0%
Resources	Resource initialization, manipulation, and release	1.3%	100.0%
Timing	Thread synchronization, races	0.3%	100.0%
Other		25.1%	16.3%

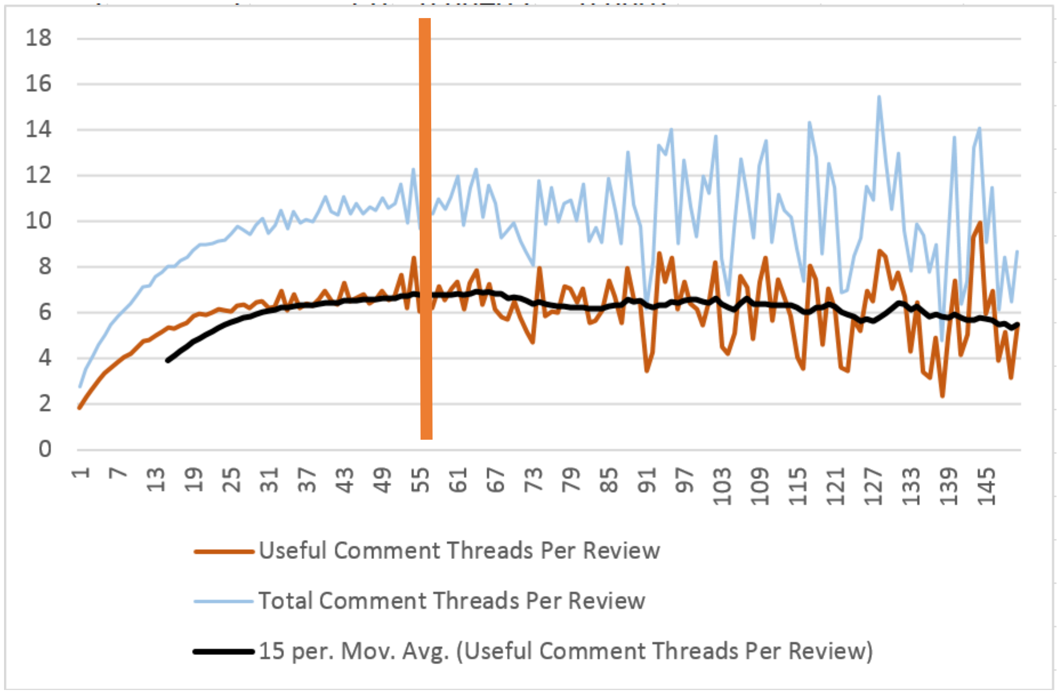
Adapted from : *M. Mäntylä and C. Lassenius. What Types of Defects Are Really Discovered in Code Reviews? IEEE Transactions Software Engineering, 35(3):430–448, 2009*

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



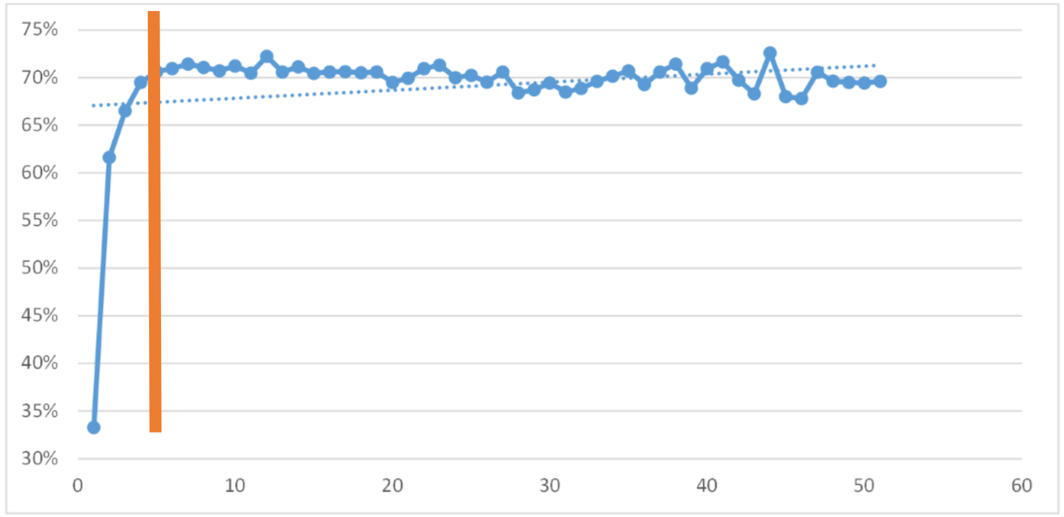
Anecdotally: smaller reviews are "better"

< ~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

By: Amiangshu Bosu (U of Alabama), Michaela Greiler (TSE), Christian Bird (Microsoft Research Redmond)

Relevant Experience Makes for Better Reviewers



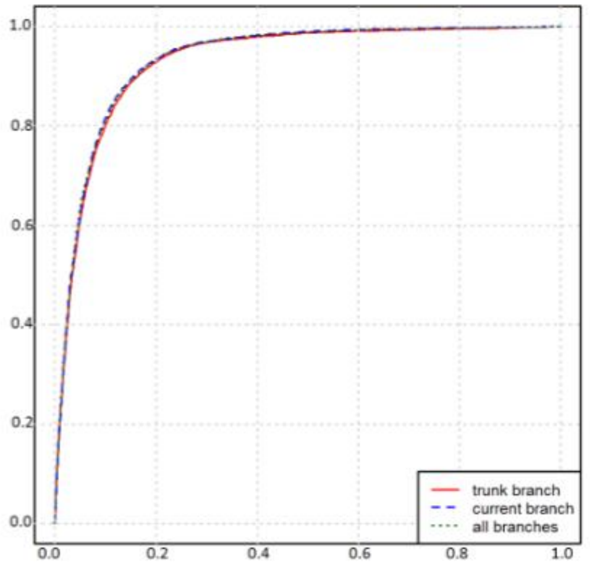
Reviewer's comment usefulness vs. number of previous reviews on a changed file

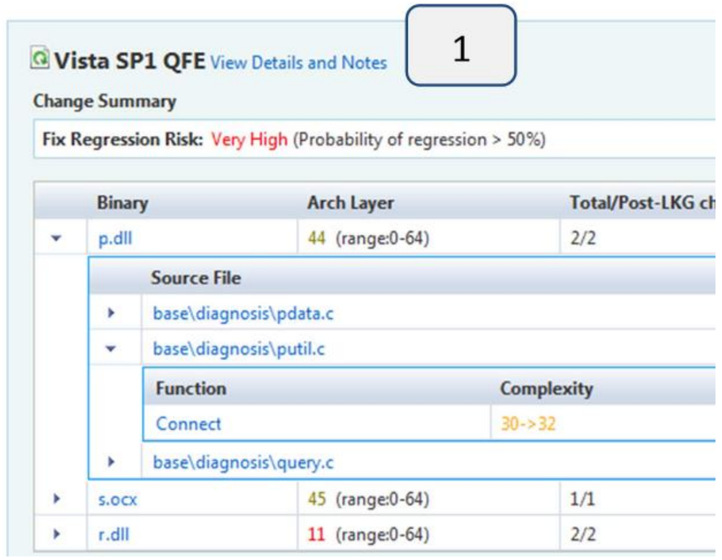
Reviewers with prior experience with the changed file produce much more useful feedback

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

By: Amiangshu Bosu (U of Alabama), Michaela Greiler (TSE), Christian Bird (Microsoft Research Redmond)

Risk of Defects In a Change Can Be Predicted





Vista SP1 QFE View Details and Notes

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
s.ocx	45 (range:0-64)	1/1
r.dll	11 (range:0-64)	2/2

Source File
base\diagnosis\pdata.c
base\diagnosis\putil.c
base\diagnosis\query.c

Function	Complexity
Connect	30->32

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

Code Reviewing: It Takes Time and Effort

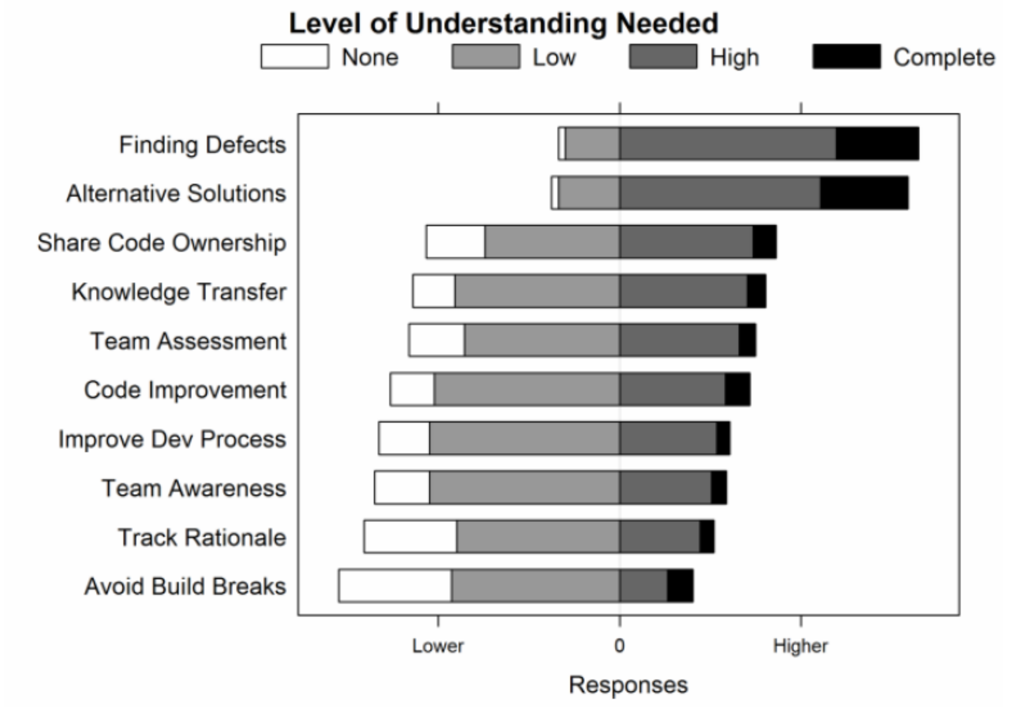


Figure 5. Developers' responses in surveys of the amount of code understanding for code review outcomes.

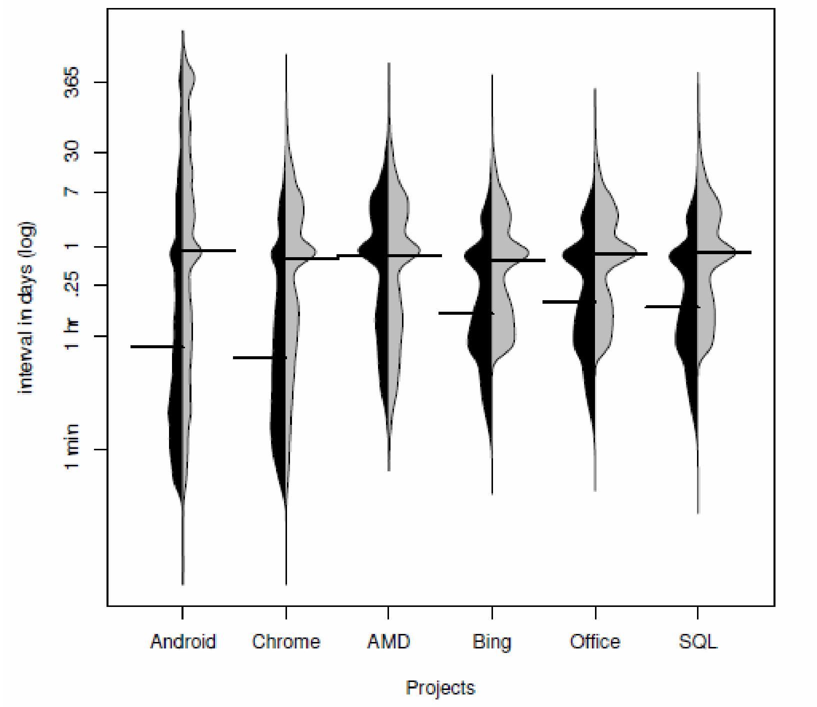


Figure 1: First Response on left (we do not have first response data for AMD) and Full interval on right

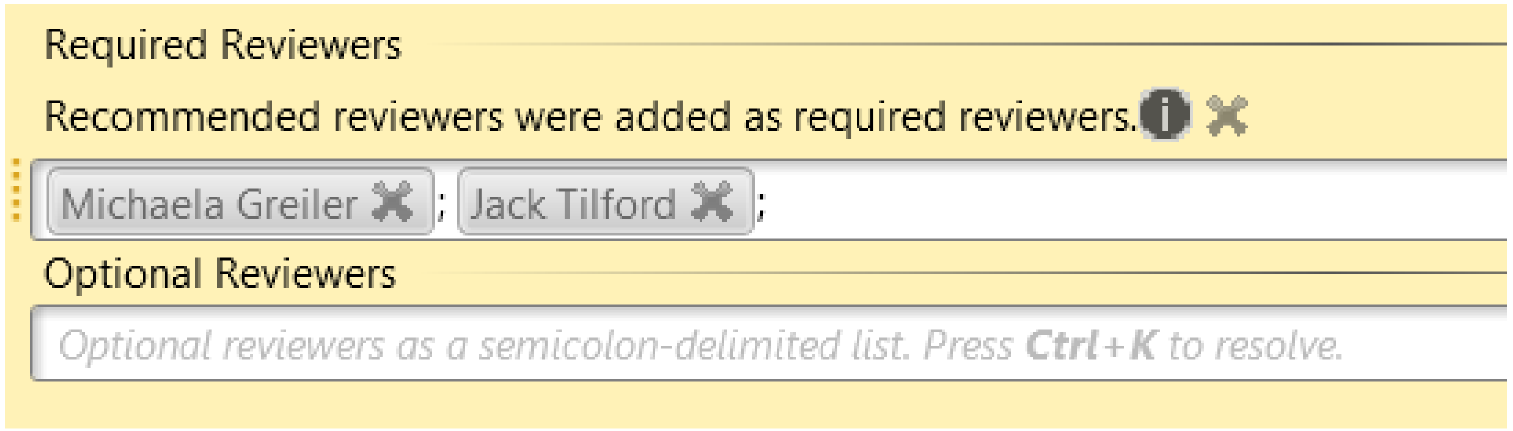
Alberto Bacchelli, Christian Bird. *Expectations, Outcomes, and Challenges Of Modern Code Review*

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

Improving the tools and the workflow

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


Required Reviewers

Recommended reviewers were added as required reviewers.

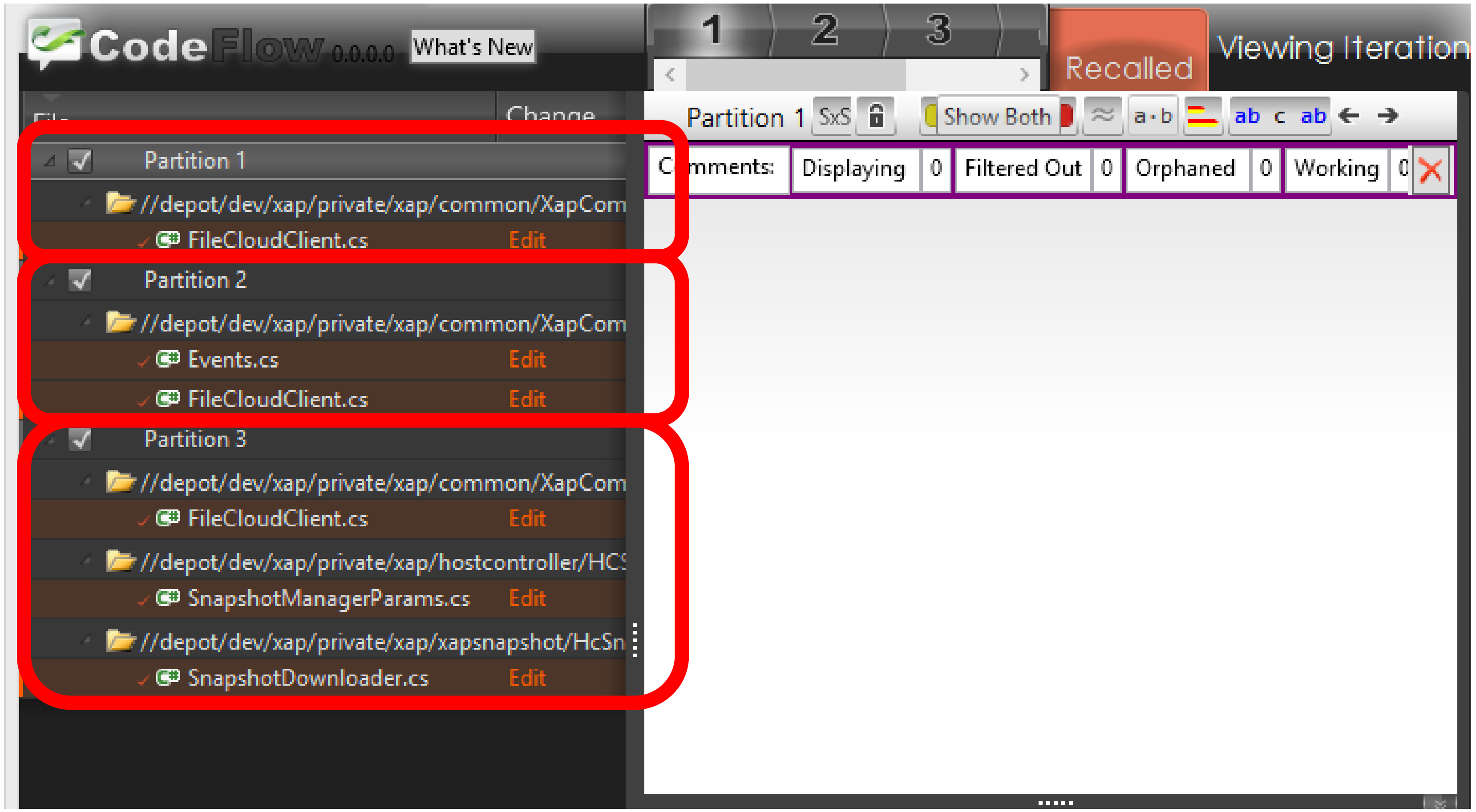
Michaela Greiler; Jack Tilford;

Optional Reviewers

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

By: Christian Bird, Birendra Acharya, Michaela Greiler, Trevor Carnahan (Microsoft Research Redmond and TSE)

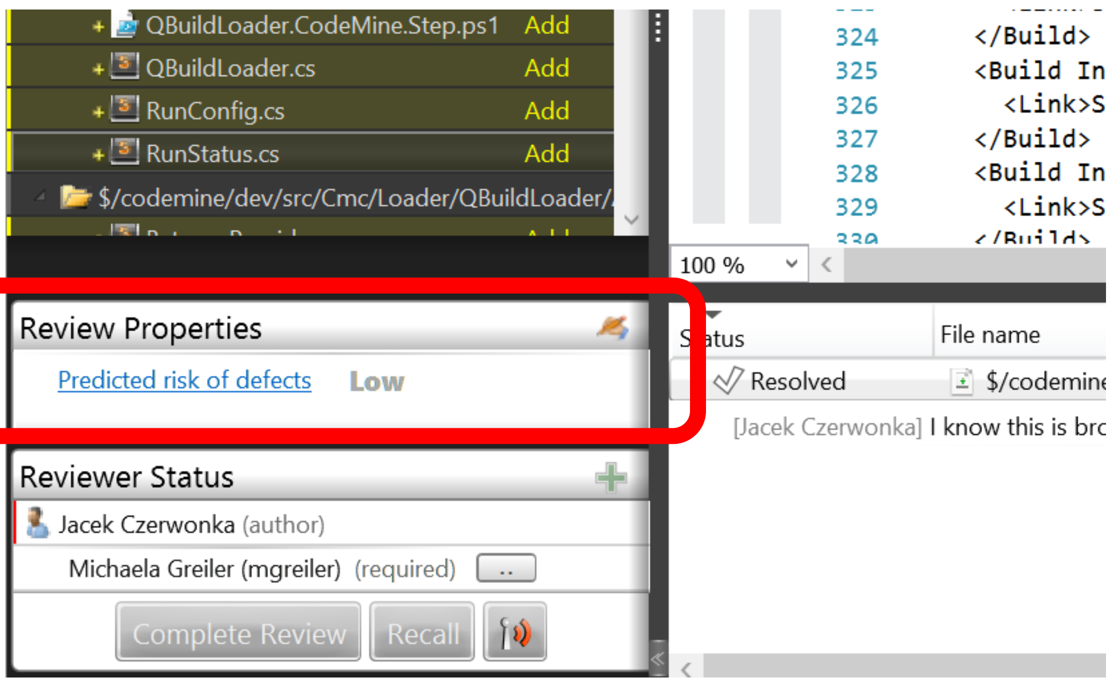
Change Decomposition



Partition 1
//depot/dev/xap/private/xap/common/XapCom
FileCloudClient.cs Edit
Partition 2
//depot/dev/xap/private/xap/common/XapCom
Events.cs Edit
FileCloudClient.cs Edit
Partition 3
//depot/dev/xap/private/xap/common/XapCom
FileCloudClient.cs Edit
//depot/dev/xap/private/xap/hostcontroller/HCS
SnapshotManagerParams.cs Edit
//depot/dev/xap/private/xap/xapsnapshot/HcSn
SnapshotDownloader.cs Edit

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

Change Risk Prediction



QBuildLoader.CodeMine.Step.ps1 Add
QBuildLoader.cs Add
RunConfig.cs Add
RunStatus.cs Add
$/codemine/dev/src/Cmc/Loader/QBuildLoader/

```
    </Build>
    <Build In
      <Link>S
    </Build>
    <Build In
      <Link>S
```

100 %

Review Properties

Predicted risk of defects Low

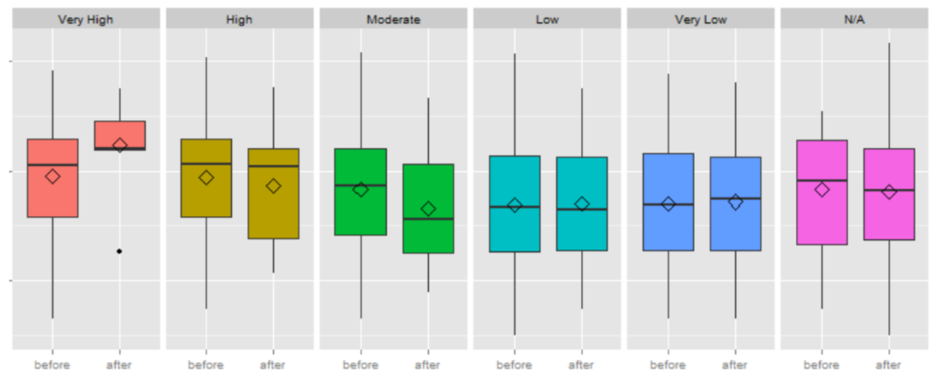
Status | File name
Resolved | $/codemine

[Jacek Czerwonka] I know this is brc

Reviewer Status

Jacek Czerwonka (author)
Michaela Greiler (mgreiler) (required)

Complete Review Recall

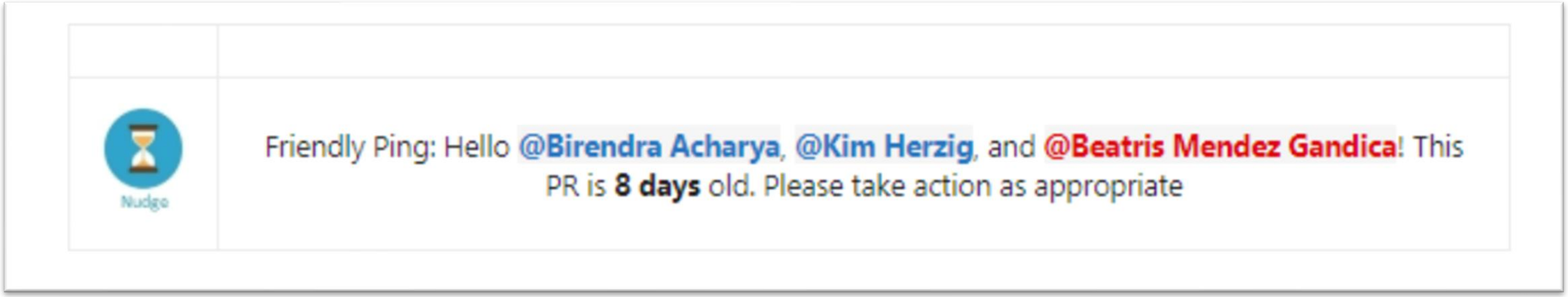


Time to first sign-off before and after enabling "change risk" feature

By: Nachi Nagappan, Jacek Czerwonka, Birendra Acharya (Microsoft Research Redmond and TSE)

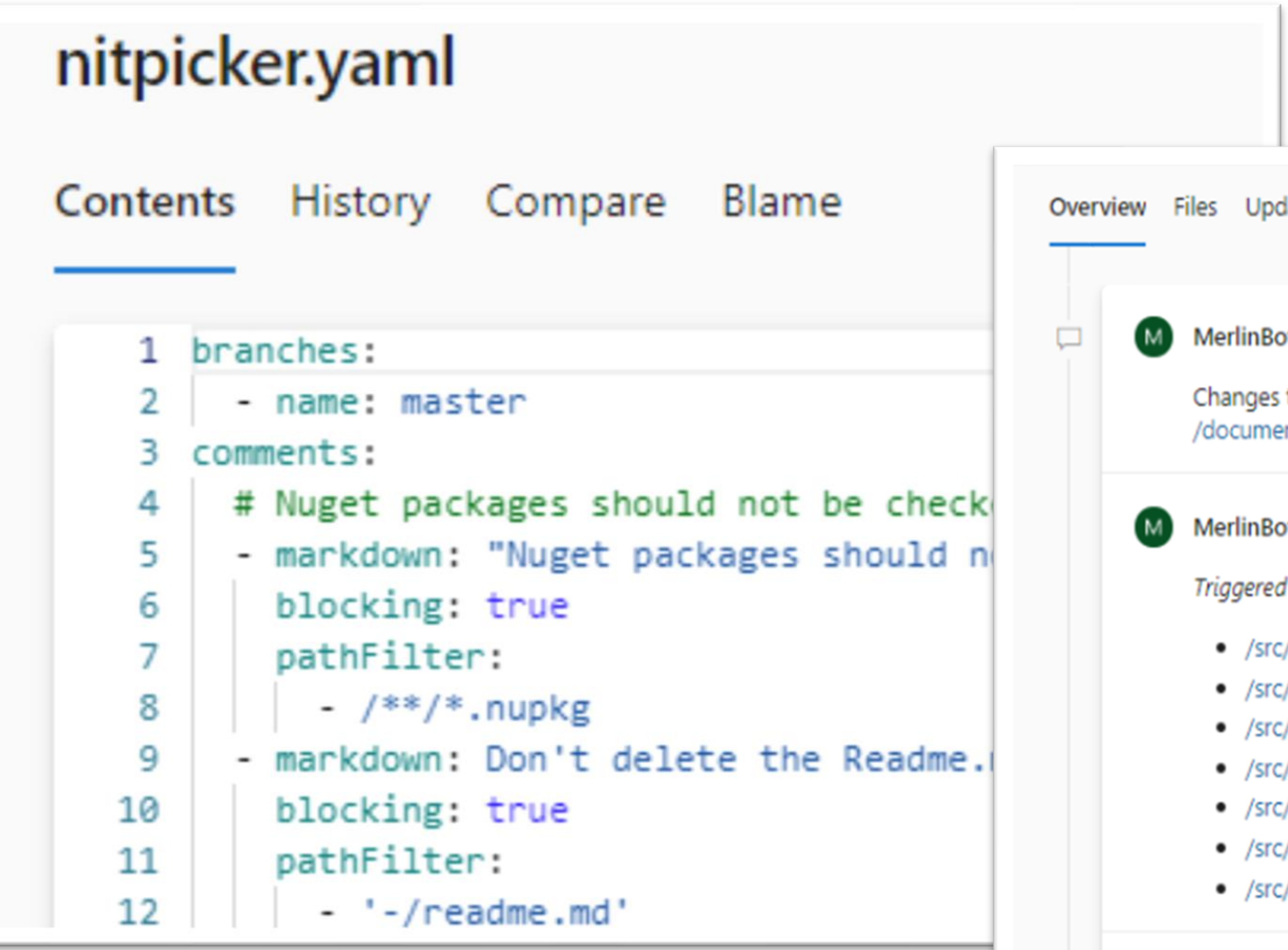
By: Kim Herzig (Microsoft Research Cambridge)

Timely Nudges



Friendly Ping: Hello @Birendra Acharya, @Kim Herzig, and @Beatris Mendez Gandica! This PR is **8 days** old. Please take action as appropriate

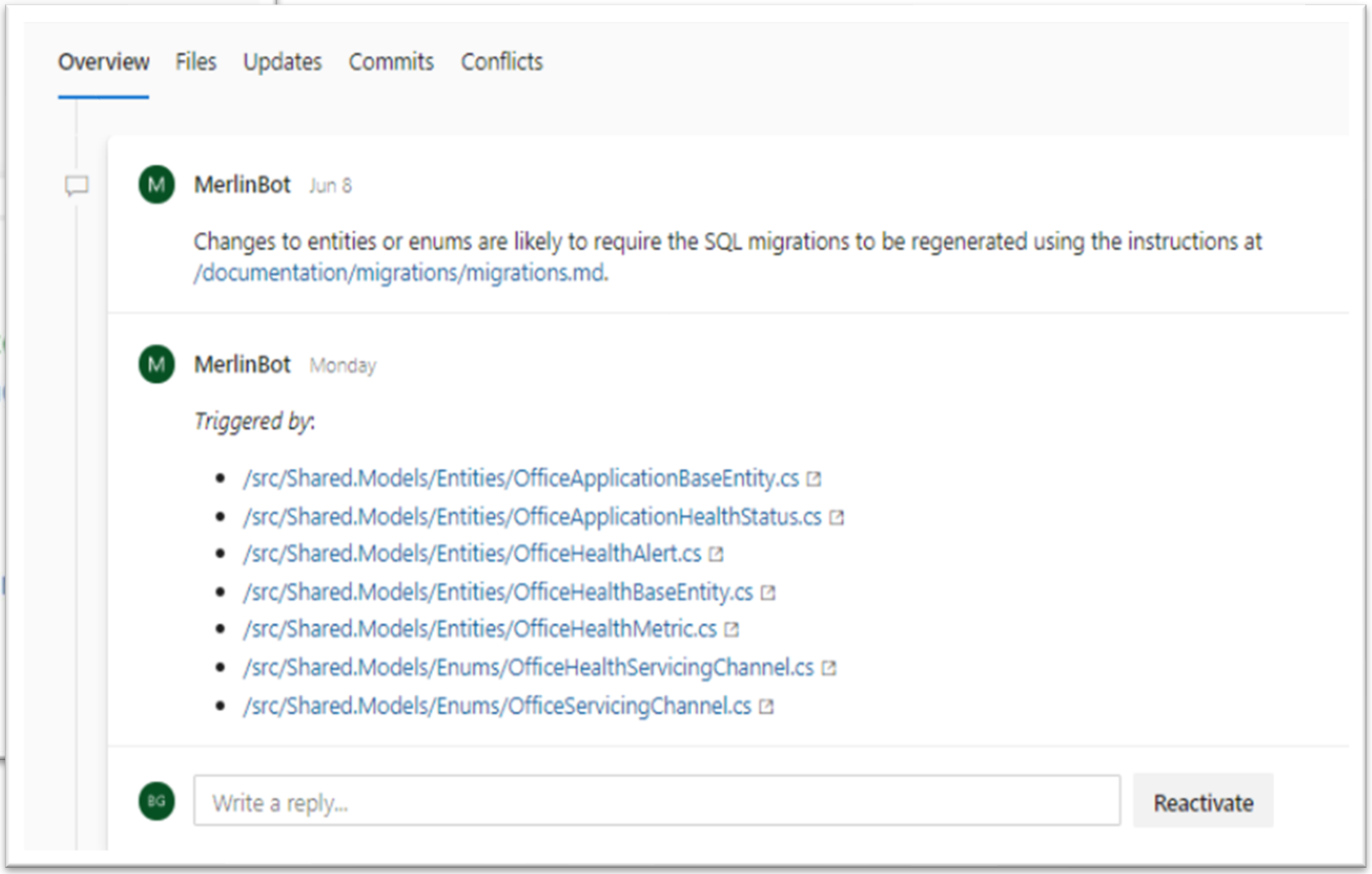
Encode Tribal Knowledge



nitpicker.yaml

Contents History Compare Blame

```yaml
branches:
  - name: master
comments:
  # Nuget packages should not be check
  - markdown: "Nuget packages should n
    blocking: true
    pathFilter:
      - /**/*.nupkg
  - markdown: Don't delete the Readme.
    blocking: true
    pathFilter:
      - '-/readme.md'
```



Overview Files Updates Commits Conflicts

MerlinBot Jun 8

Changes to entities or enums are likely to require the SQL migrations to be regenerated using the instructions at /documentation/migrations/migrations.md.

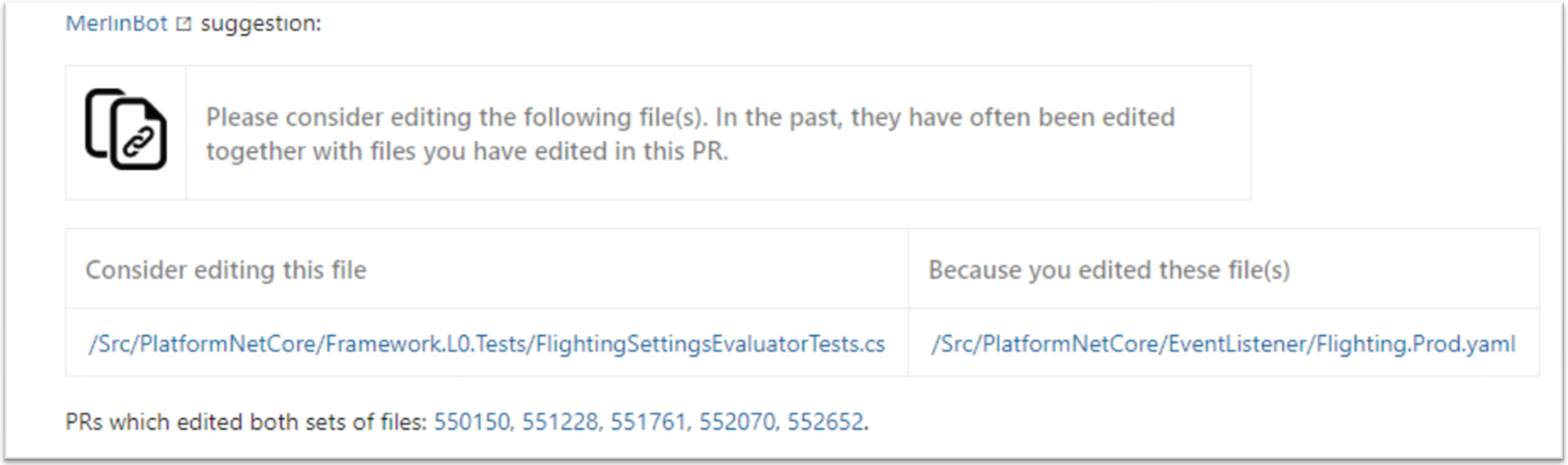
MerlinBot Monday

Triggered by:

- /src/Shared.Models/Entities/OfficeApplicationBaseEntity.cs
- /src/Shared.Models/Entities/OfficeApplicationHealthStatus.cs
- /src/Shared.Models/Entities/OfficeHealthAlert.cs
- /src/Shared.Models/Entities/OfficeHealthBaseEntity.cs
- /src/Shared.Models/Entities/OfficeHealthMetric.cs
- /src/Shared.Models/Enums/OfficeHealthServicingChannel.cs
- /src/Shared.Models/Enums/OfficeServicingChannel.cs

Write a reply... Reactivate

Discover Past Patterns



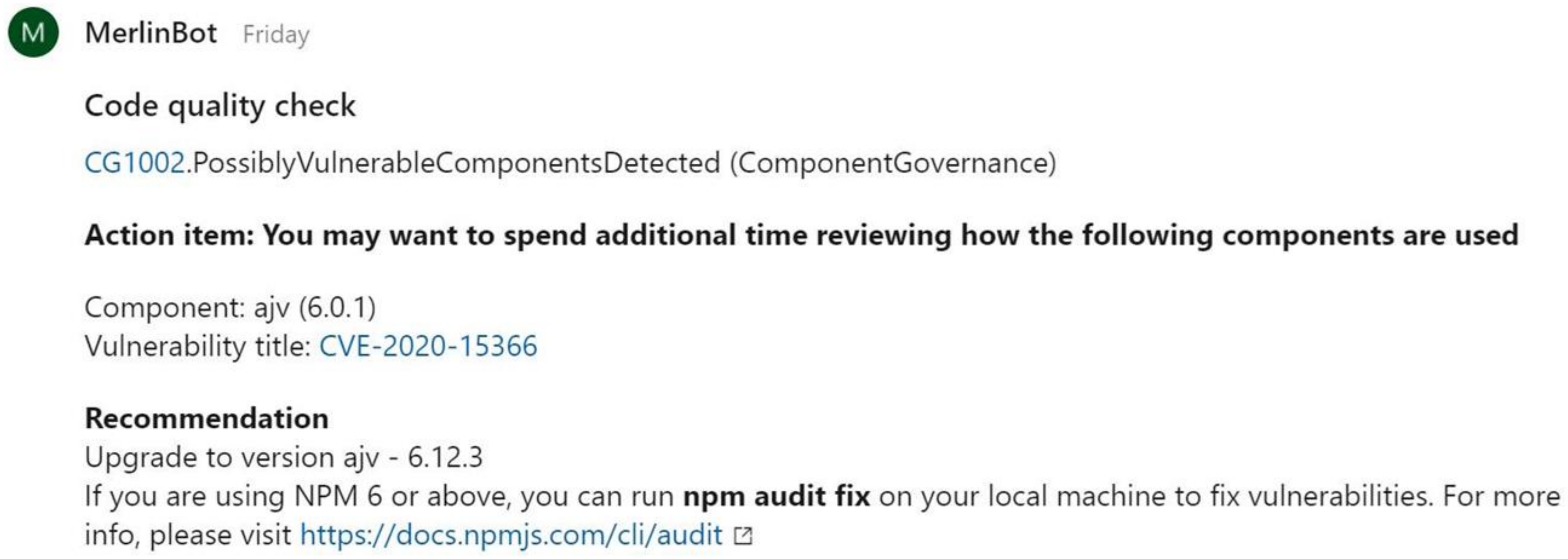
MerlinBot suggestion:

Please consider editing the following file(s). In the past, they have often been edited together with files you have edited in this PR.

Consider editing this file	Because you edited these file(s)
/Src/PlatformNetCore/Framework.L0.Tests/FlightingSettingsEvaluatorTests.cs	/Src/PlatformNetCore/EventListener/Flighting.Prod.yaml

PRs which edited both sets of files: 550150, 551228, 551761, 552070, 552652.

Propose (and Create) Fixes



M MerlinBot Friday

Code quality check

CG1002.PossiblyVulnerableComponentsDetected (ComponentGovernance)

Action item: You may want to spend additional time reviewing how the following components are used

Component: ajv (6.0.1)
Vulnerability title: CVE-2020-15366

Recommendation
Upgrade to version ajv - 6.12.3
If you are using NPM 6 or above, you can run **npm audit fix** on your local machine to fix vulnerabilities. For more info, please visit https://docs.npmjs.com/cli/audit

Criteria

- **Usable**: Does it work as advertised?
- **Beneficial:** What improvement this creates vs. current process?
- **Generalizable**: Does it apply to all teams?
- **Scalable**: Does it scale to large teams / code bases?
- **Cost-effective**: What are implementation and *maintenance* costs?
- **Fitness:** Does it extend an existing workflow or creates a new one?
- **Actionable**: Does the user know what to do?
- **Bounded**: What maximum damage this can create?

Further reading

- Code Reviewing in the Trenches: Understanding Challenges, Best Practices and Tool Needs - Microsoft Research