FOSDEM 2025

Using LLMs to support Firefox developers

Marco Castelluccio

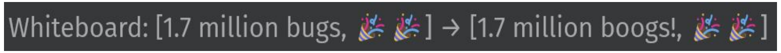
Mozilla

The Firefox Scale

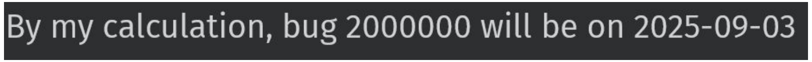
- Hundreds of bug reports and feature requests per day
  - **1’944’000 bug reports** at this time!

- We release **every 4 weeks** with thousands of changes

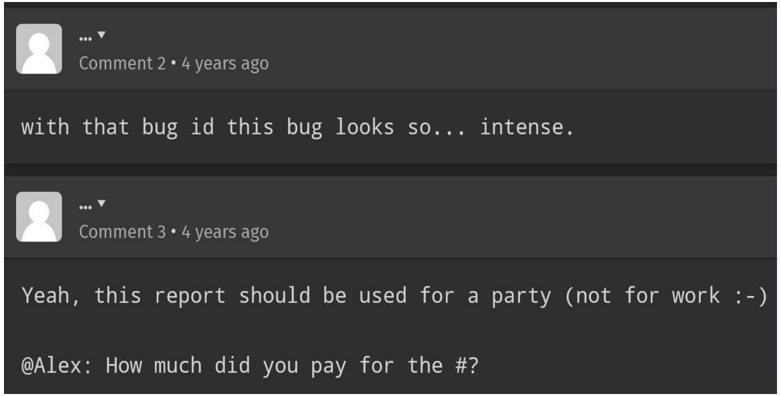
- **12 major releases** during 2024



Whiteboard: [1.7 million bugs, 🎉 🎉] → [1.7 million boogs!, 🎉 🎉]



By my calculation, bug 2000000 will be on 2025-09-03



… ▾
Comment 2 • 4 years ago

```
with that bug id this bug looks so... intense.
```

… ▾
Comment 3 • 4 years ago

```
Yeah, this report should be used for a party (not for work :-)

@Alex: How much did you pay for the #?
```

The Firefox Scale

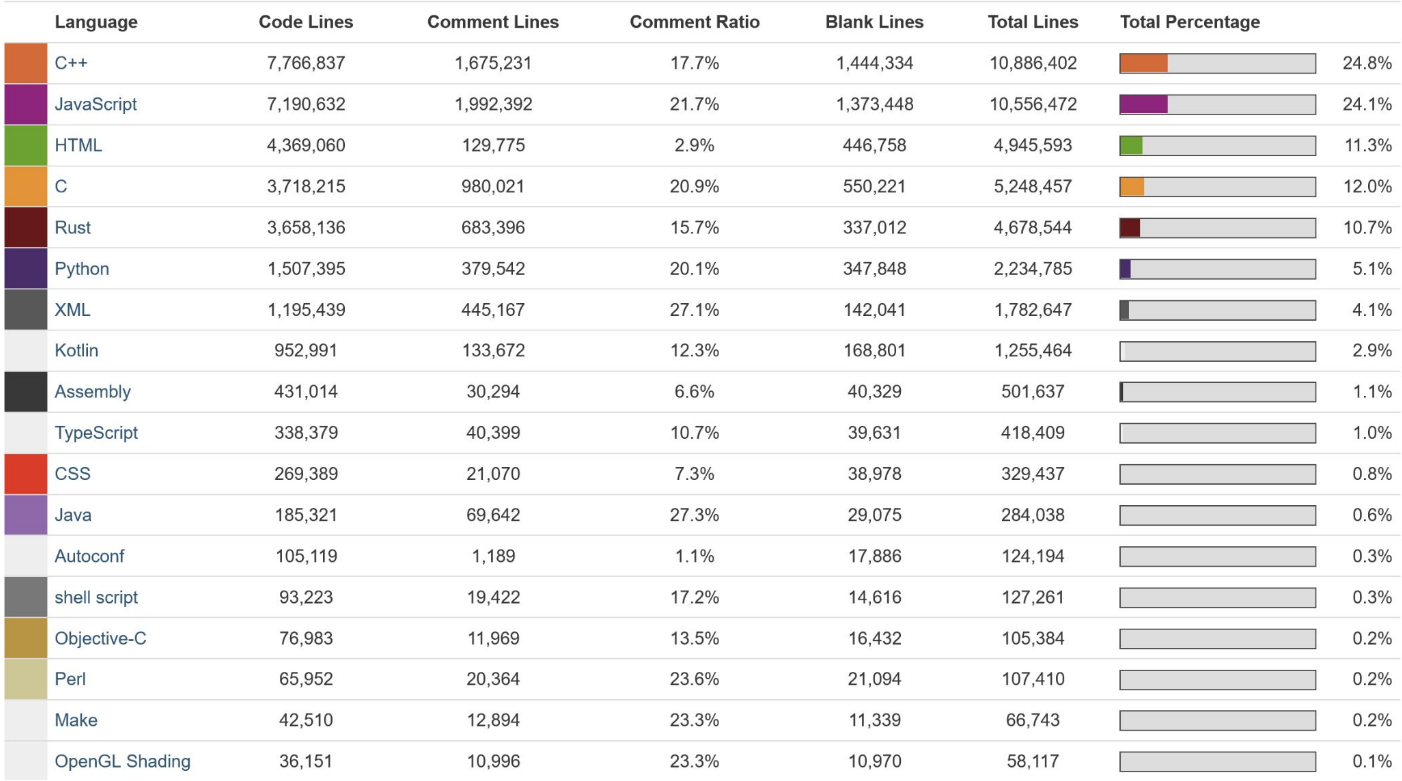
- One of the biggest and most complex software
  - With legacy and tech debt (Netscape was open sourced **27 years ago**, we celebrated 20 years of Firefox in November)

- **950’000 commits** made by **10’000 contributors** representing **32’000’000 lines of code**

- **46’000 commits last year**

- **1’100 unique contributors last year**

The Firefox Scale



Language	Code Lines	Comment Lines	Comment Ratio	Blank Lines	Total Lines	Total Percentage
C++	7,766,837	1,675,231	17.7%	1,444,334	10,886,402	24.8%
JavaScript	7,190,632	1,992,392	21.7%	1,373,448	10,556,472	24.1%
HTML	4,369,060	129,775	2.9%	446,758	4,945,593	11.3%
C	3,718,215	980,021	20.9%	550,221	5,248,457	12.0%
Rust	3,658,136	683,396	15.7%	337,012	4,678,544	10.7%
Python	1,507,395	379,542	20.1%	347,848	2,234,785	5.1%
XML	1,195,439	445,167	27.1%	142,041	1,782,647	4.1%
Kotlin	952,991	133,672	12.3%	168,801	1,255,464	2.9%
Assembly	431,014	30,294	6.6%	40,329	501,637	1.1%
TypeScript	338,379	40,399	10.7%	39,631	418,409	1.0%
CSS	269,389	21,070	7.3%	38,978	329,437	0.8%
Java	185,321	69,642	27.3%	29,075	284,038	0.6%
Autoconf	105,119	1,189	1.1%	17,886	124,194	0.3%
shell script	93,223	19,422	17.2%	14,616	127,261	0.3%
Objective-C	76,983	11,969	13.5%	16,432	105,384	0.2%
Perl	65,952	20,364	23.6%	21,094	107,410	0.2%
Make	42,510	12,894	23.3%	11,339	66,743	0.2%
OpenGL Shading	36,151	10,996	23.3%	10,970	58,117	0.1%

The Idea

- ~~Can we automate reviews?~~
- ~~Can we make reviews faster?~~
- Can we identify more issues and improve code quality through code review by supporting developers in the process?

Collaboration

- Mozilla
- Ubisoft
- Queen's School of Computing
- University of Montreal

- All working in an open source project: https://github.com/mozilla/bugbug

- Paper WIP

Agenda

→ **01 A trip down prompt engineering lane**

02 What it looks like

03 Do people like it?

04 Next steps

05 What else?

A trip down prompt engineering lane

Start with a simple prompt.

Please find issues in the following patch: {PATCH}

A trip down prompt engineering lane

Start with a simple prompt.

Please find issues in the following patch: {PATCH}

Add “human” examples.

Here are some examples of past comments:

A trip down prompt engineering lane

Start with a simple prompt.

Please find issues in the following patch: {PATCH}

Fight against the urge to be nice.

Filter out comments that are praising (example: ”This is a good addition to the code.”).

Add “human” examples.

Here are some examples of past comments:

A trip down prompt engineering lane

Start with a simple prompt.

Please find issues in the following patch: {PATCH}

Add “human” examples.

Here are some examples of past comments:

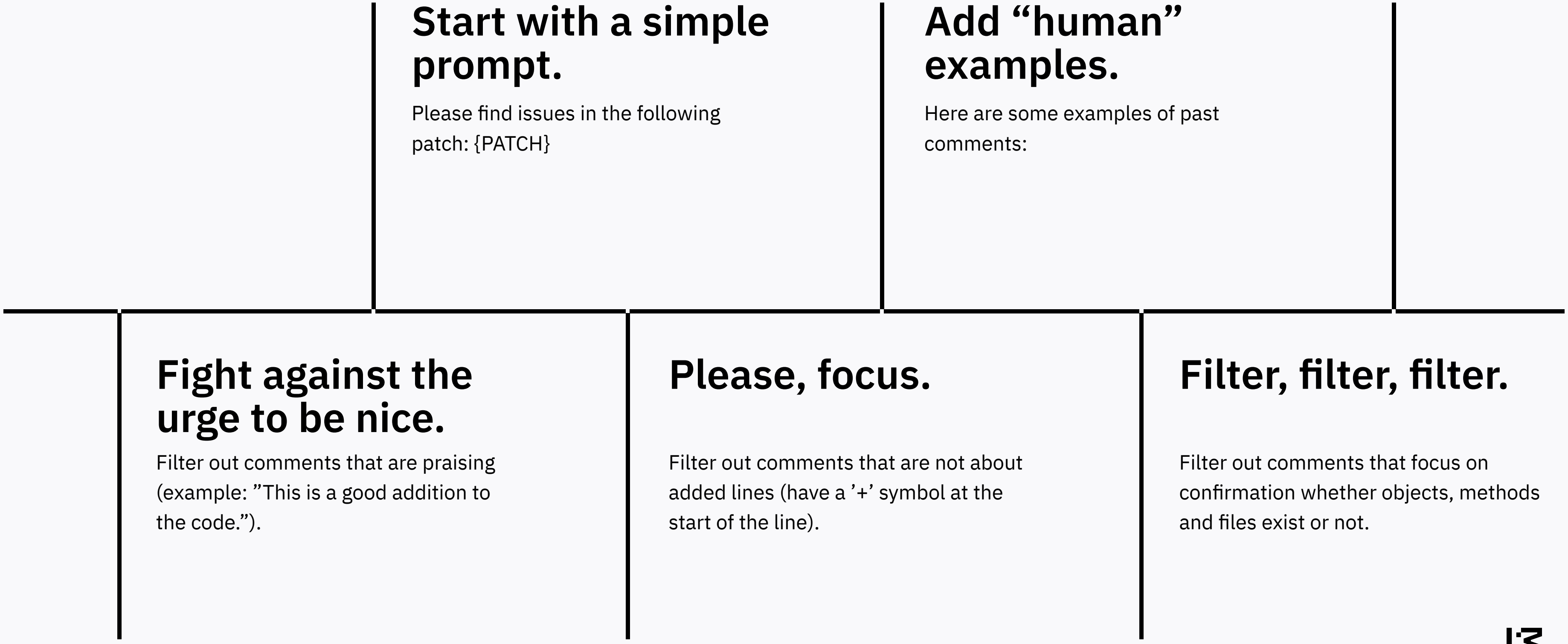
Fight against the urge to be nice.

Filter out comments that are praising (example: ”This is a good addition to the code.”).

Please, focus.

Filter out comments that are not about added lines (have a ’+’ symbol at the start of the line).

A trip down prompt engineering lane



Fight against the urge to be nice.

Filter out comments that are praising (example: ”This is a good addition to the code.”).

Start with a simple prompt.

Please find issues in the following patch: {PATCH}

Please, focus.

Filter out comments that are not about added lines (have a ’+’ symbol at the start of the line).

Add “human” examples.

Here are some examples of past comments:

Filter, filter, filter.

Filter out comments that focus on confirmation whether objects, methods and files exist or not.

2nd trip down prompt engineering lane

Summary first.

Explain the patch, then review.

2nd trip down prompt engineering lane

Summary first.

Explain the patch, then review.

More context?

Do you need to see more context around changed lines?

2nd trip down prompt engineering lane

Summary first.

Explain the patch, then review.

More context?

Do you need to see more context around changed lines?

Search code?

Do you need to see the definition/implementation of the functions used within the patch?

2nd trip down prompt engineering lane

Search code?

Do you need to see the definition/implementation of the functions used within the patch?

Summary first.

Explain the patch, then review.

Add “bad” examples.

Overall, the patch seems to be well implemented with no major concerns.

More context?

Do you need to see more context around changed lines?

Agenda

01 A trip down prompt engineering lane

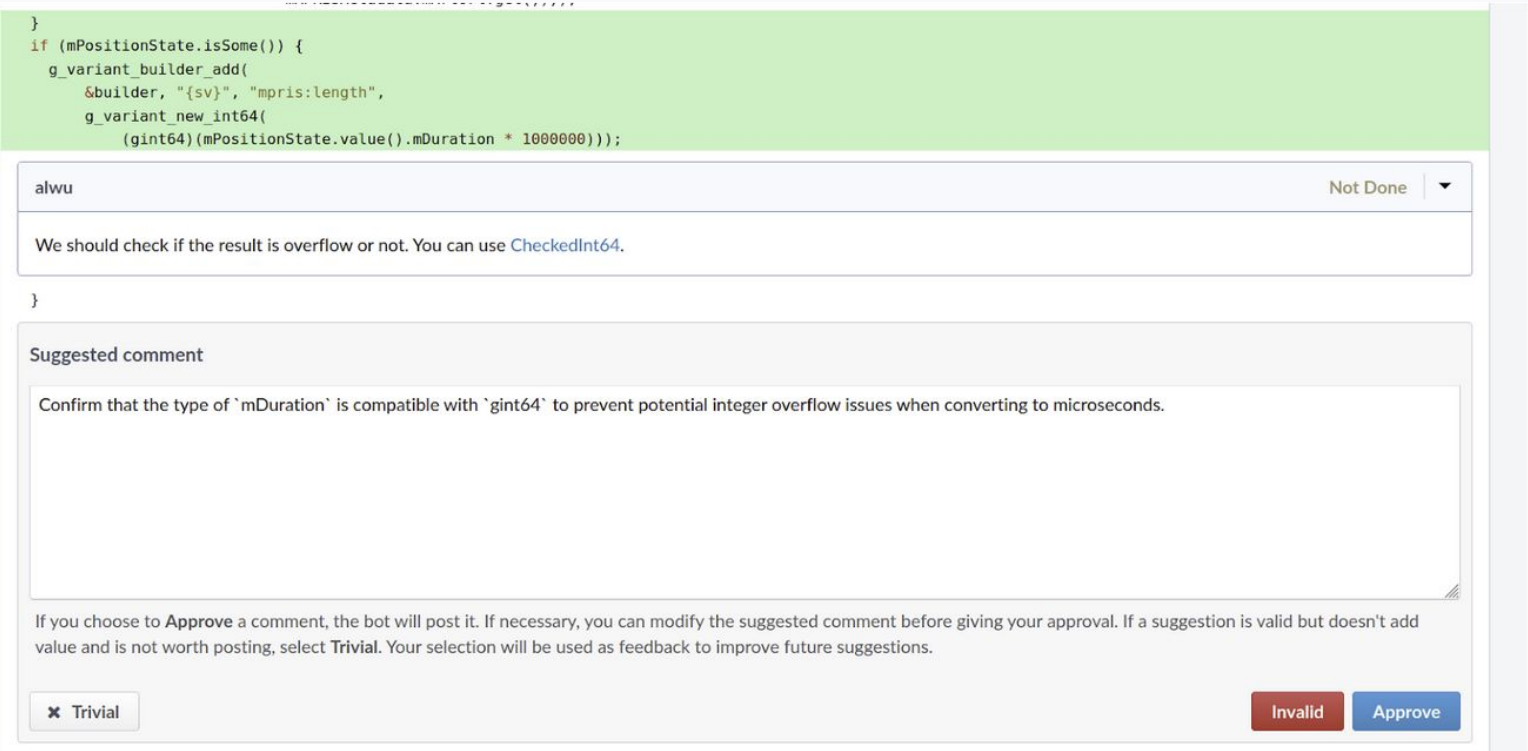
→ 02 What it looks like

03 Do people like it?

04 Next steps

05 What else?

What it looks like



```
  }
  if (mPositionState.isSome()) {
    g_variant_builder_add(
        &builder, "{sv}", "mpris:length",
        g_variant_new_int64(
            (gint64)(mPositionState.value().mDuration * 1000000)));
```

alwu

We should check if the result is overflow or not. You can use CheckedInt64.

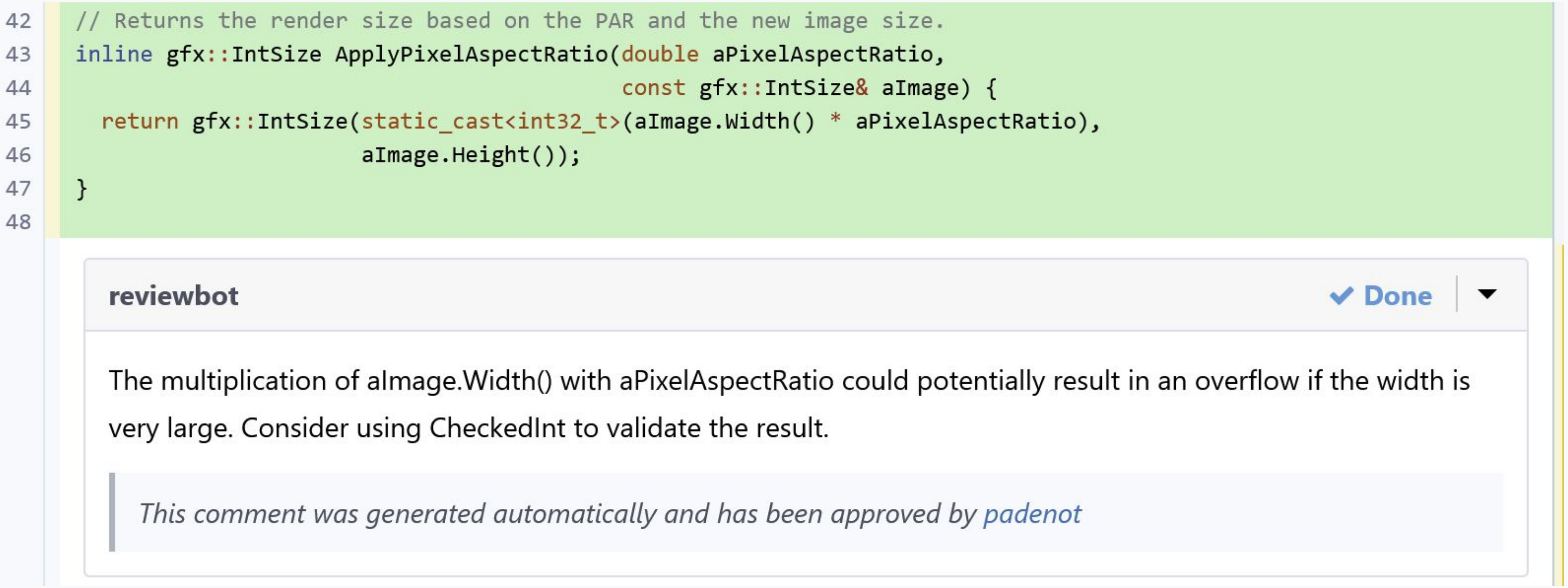
```
  }
```

Suggested comment

Confirm that the type of `mDuration` is compatible with `gint64` to prevent potential integer overflow issues when converting to microseconds.

If you choose to **Approve** a comment, the bot will post it. If necessary, you can modify the suggested comment before giving your approval. If a suggestion is valid but doesn't add value and is not worth posting, select **Trivial**. Your selection will be used as feedback to improve future suggestions.

What it looks like



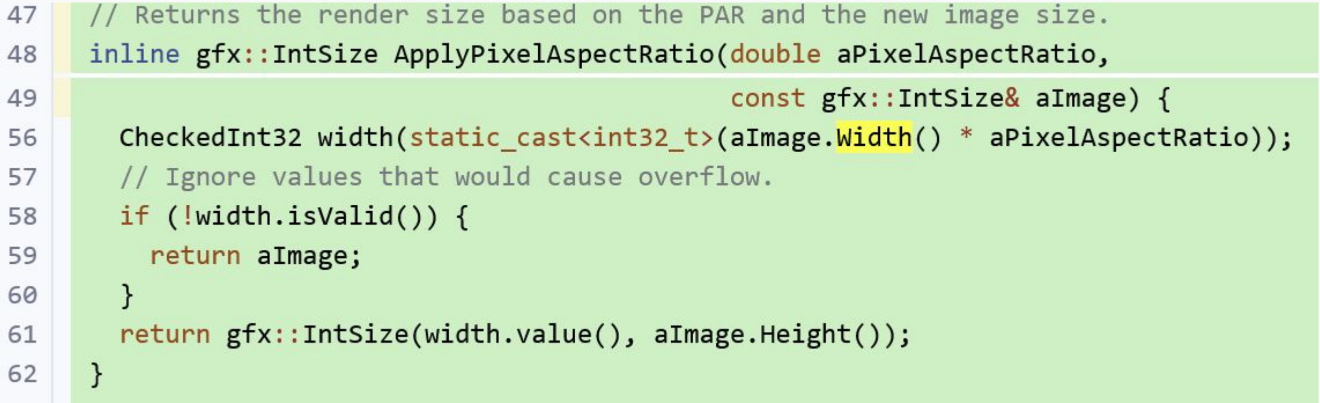
```
// Returns the render size based on the PAR and the new image size.
inline gfx::IntSize ApplyPixelAspectRatio(double aPixelAspectRatio,
                                          const gfx::IntSize& aImage) {
  return gfx::IntSize(static_cast<int32_t>(aImage.Width() * aPixelAspectRatio),
                      aImage.Height());
}

```

reviewbot

The multiplication of aImage.Width() with aPixelAspectRatio could potentially result in an overflow if the width is very large. Consider using CheckedInt to validate the result.

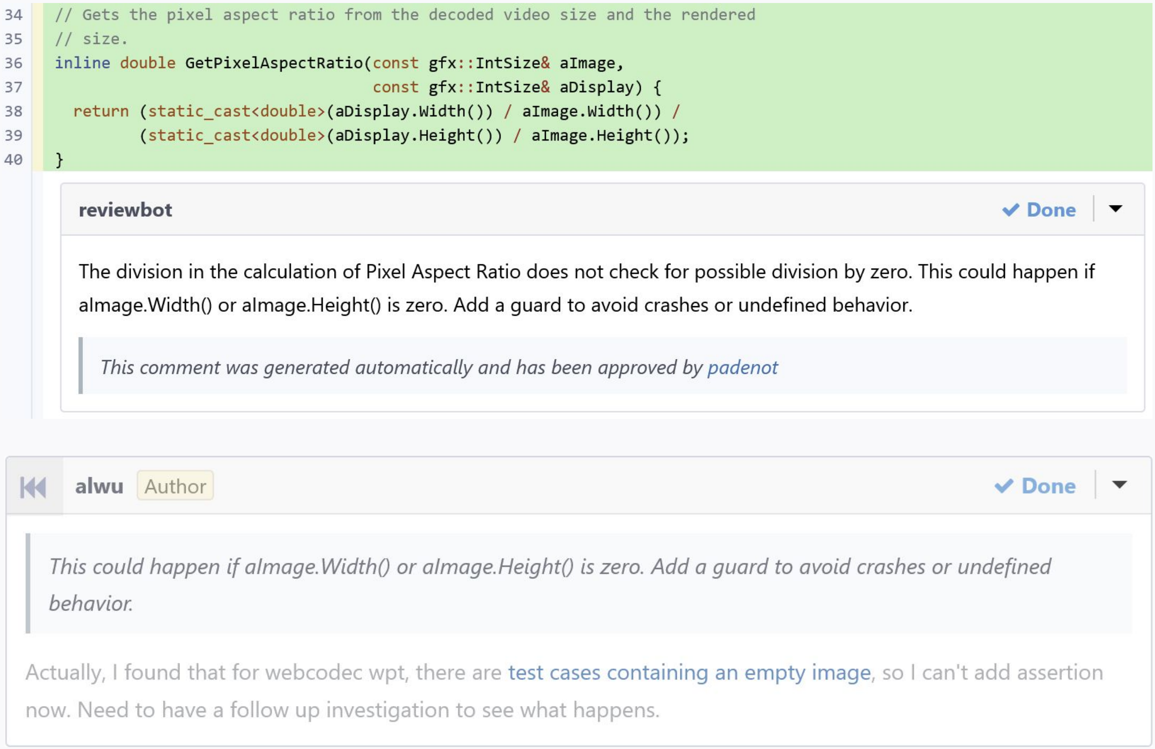
> *This comment was generated automatically and has been approved by padenot*

What it looks like



```
// Returns the render size based on the PAR and the new image size.
inline gfx::IntSize ApplyPixelAspectRatio(double aPixelAspectRatio,
                                          const gfx::IntSize& aImage) {
  CheckedInt32 width(static_cast<int32_t>(aImage.Width() * aPixelAspectRatio));
  // Ignore values that would cause overflow.
  if (!width.isValid()) {
    return aImage;
  }
  return gfx::IntSize(width.value(), aImage.Height());
}
```

What it looks like



```
// Gets the pixel aspect ratio from the decoded video size and the rendered
// size.
inline double GetPixelAspectRatio(const gfx::IntSize& aImage,
                                  const gfx::IntSize& aDisplay) {
  return (static_cast<double>(aDisplay.Width()) / aImage.Width()) /
         (static_cast<double>(aDisplay.Height()) / aImage.Height());
}
```

reviewbot

The division in the calculation of Pixel Aspect Ratio does not check for possible division by zero. This could happen if aImage.Width() or aImage.Height() is zero. Add a guard to avoid crashes or undefined behavior.

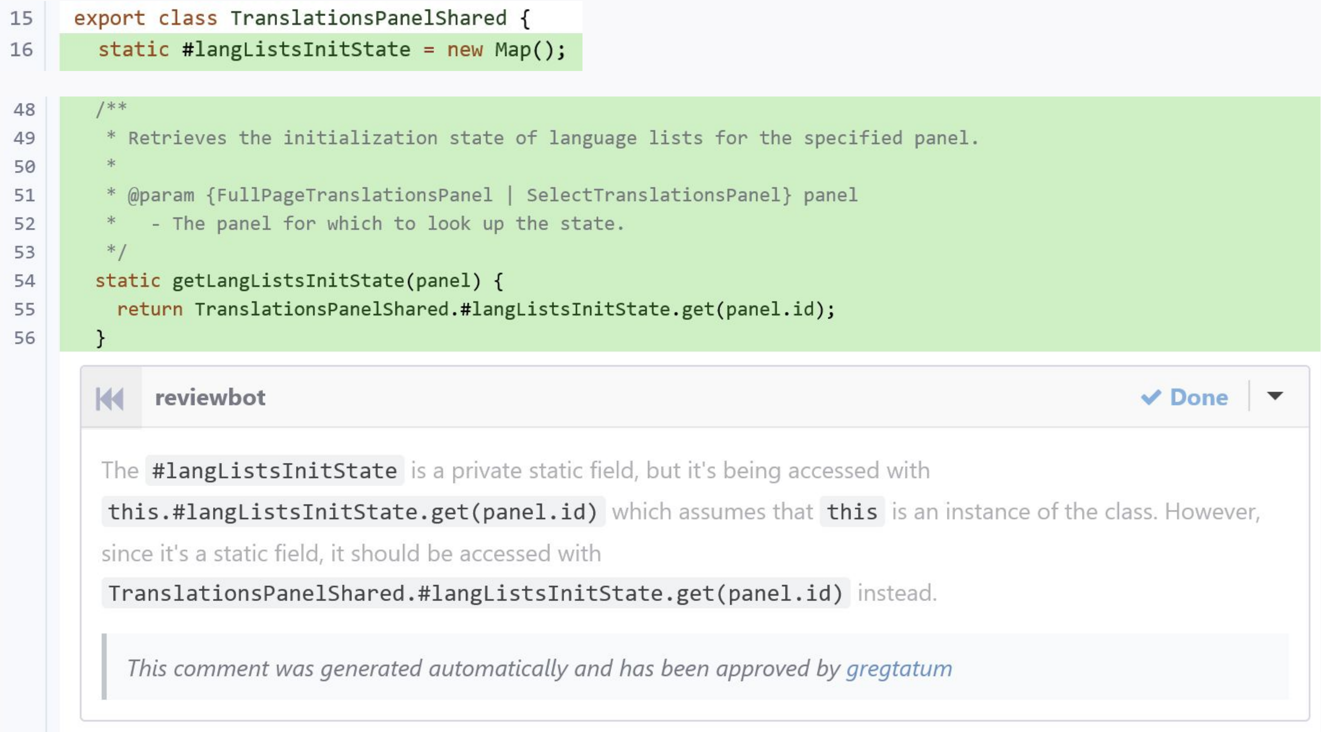
> *This comment was generated automatically and has been approved by padenot*

alwu Author

> *This could happen if aImage.Width() or aImage.Height() is zero. Add a guard to avoid crashes or undefined behavior.*

Actually, I found that for webcodec wpt, there are test cases containing an empty image, so I can't add assertion now. Need to have a follow up investigation to see what happens.

What it looks like



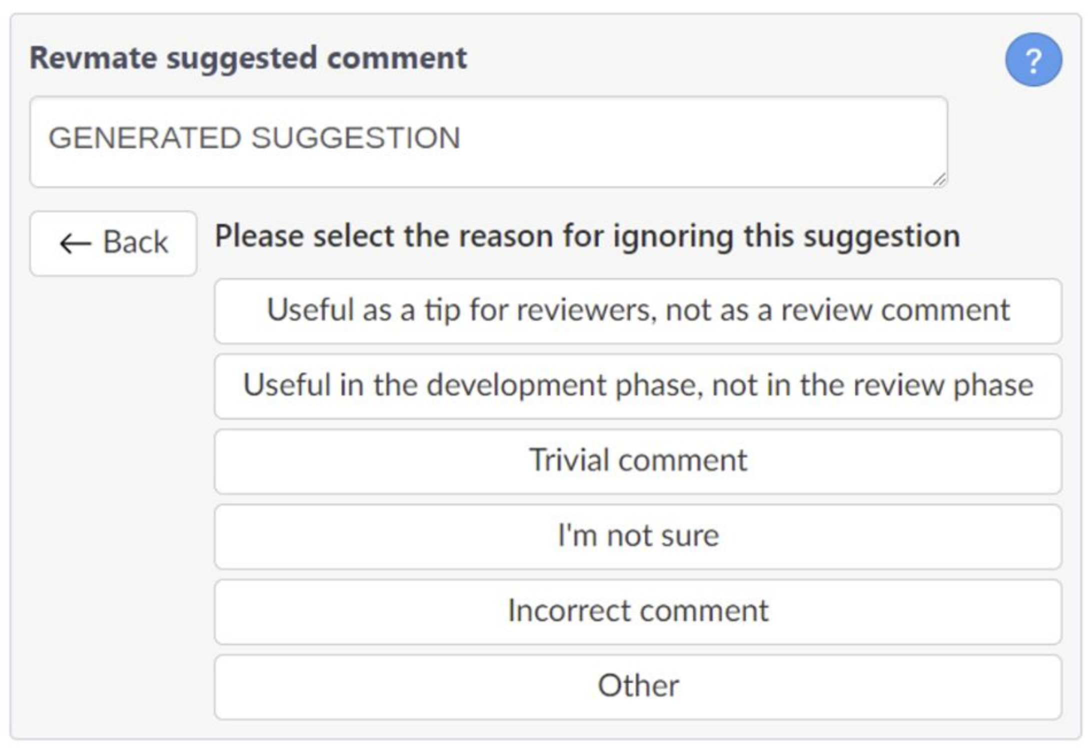
```
export class TranslationsPanelShared {
  static #langListsInitState = new Map();
```

```
  /**
   * Retrieves the initialization state of language lists for the specified panel.
   *
   * @param {FullPageTranslationsPanel | SelectTranslationsPanel} panel
   *   - The panel for which to look up the state.
   */
  static getLangListsInitState(panel) {
    return TranslationsPanelShared.#langListsInitState.get(panel.id);
  }
```

The `#langListsInitState` is a private static field, but it's being accessed with `this.#langListsInitState.get(panel.id)` which assumes that `this` is an instance of the class. However, since it's a static field, it should be accessed with `TranslationsPanelShared.#langListsInitState.get(panel.id)` instead.

> *This comment was generated automatically and has been approved by gregtatum*

What it looks like



Revmate suggested comment

GENERATED SUGGESTION

Please select the reason for ignoring this suggestion

- Useful as a tip for reviewers, not as a review comment
- Useful in the development phase, not in the review phase
- Trivial comment
- I'm not sure
- Incorrect comment
- Other

Agenda

01 A trip down prompt engineering lane

02 What it looks like

→ 03 Do people like it?

04 Next steps

05 What else?

How many comments are accepted?

8.1%

At Mozilla

7.2%

At Ubisoft

> It detects perfectly possible flaws in the code, but most of the suggestions are more in the development realm than the review.

Anonymous
Developer

- Useful as a tip for reviewers, not as a review comment
- Useful in the development phase, not in the review phase

How many comments are appreciated?

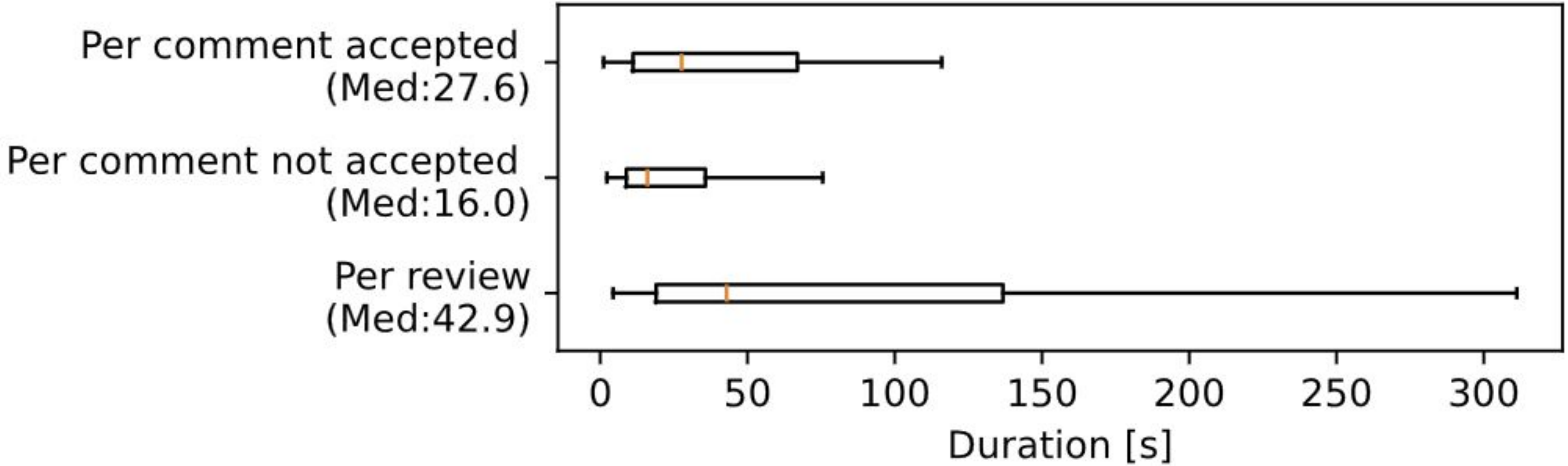
23%
At Mozilla

28.3%
At Ubisoft

Often looking at the suggestion helps me think more about the code and reflect a bit more on what is done and why it is done. So overall, I think it is a bit slower to do reviews, but my reviews are actually better now that I use the tool.

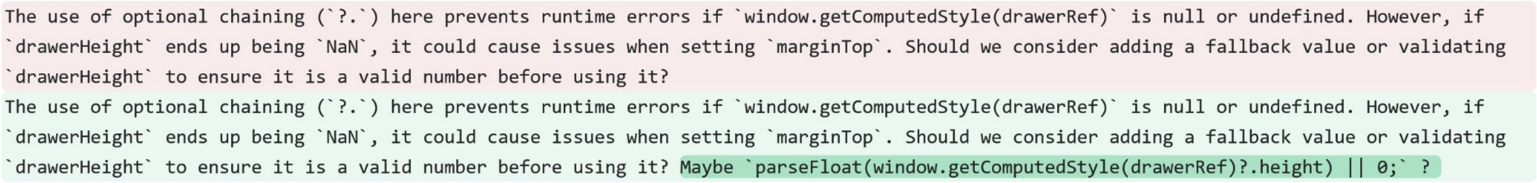
Anonymous
Developer

Does it take longer to review?



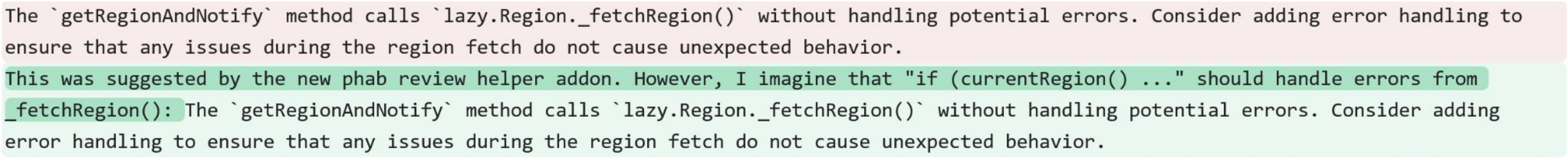
Are comments edited?

- Most comments are accepted as-is, or shortened (LLMs can be too verbose)
- A couple of examples of lengthened:



The use of optional chaining (`?.`) here prevents runtime errors if `window.getComputedStyle(drawerRef)` is null or undefined. However, if `drawerHeight` ends up being `NaN`, it could cause issues when setting `marginTop`. Should we consider adding a fallback value or validating `drawerHeight` to ensure it is a valid number before using it?

The use of optional chaining (`?.`) here prevents runtime errors if `window.getComputedStyle(drawerRef)` is null or undefined. However, if `drawerHeight` ends up being `NaN`, it could cause issues when setting `marginTop`. Should we consider adding a fallback value or validating `drawerHeight` to ensure it is a valid number before using it? Maybe `parseFloat(window.getComputedStyle(drawerRef)?.height) || 0;` ?



The `getRegionAndNotify` method calls `lazy.Region._fetchRegion()` without handling potential errors. Consider adding error handling to ensure that any issues during the region fetch do not cause unexpected behavior.

This was suggested by the new phab review helper addon. However, I imagine that "if (currentRegion() ..." should handle errors from _fetchRegion(): The `getRegionAndNotify` method calls `lazy.Region._fetchRegion()` without handling potential errors. Consider adding error handling to ensure that any issues during the region fetch do not cause unexpected behavior.

Do comments lead to changes?

74%

Of LLM comments

73%

Of Human Comments

Agenda

01 A trip down prompt engineering lane

02 What it looks like

03 Do people like it?

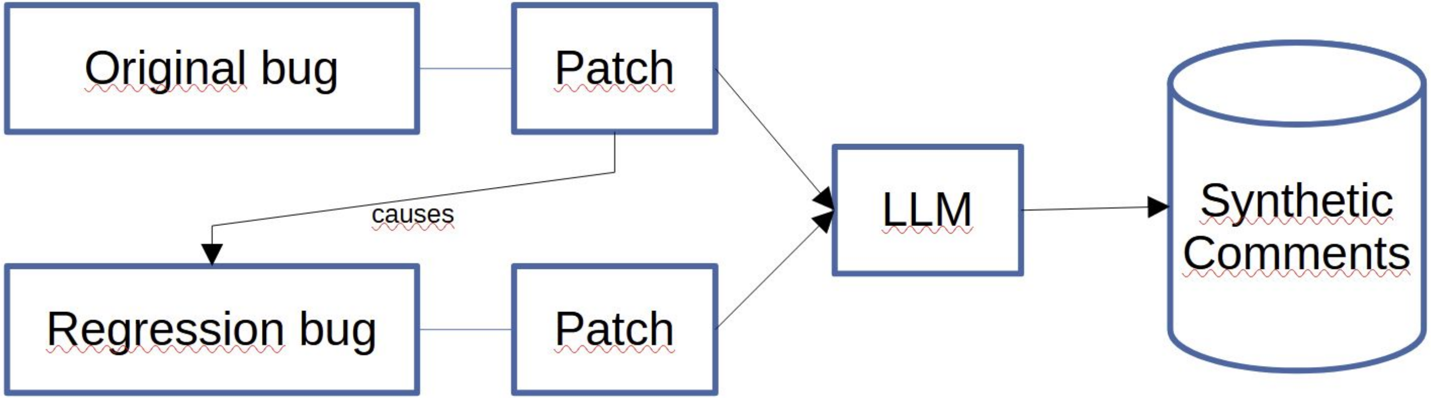
→ 04 Next steps

05 What else?

Improving bug detection capabilities

- **Regressed By** field on Bugzilla
  - Allows developers to specify which patch caused a given bug

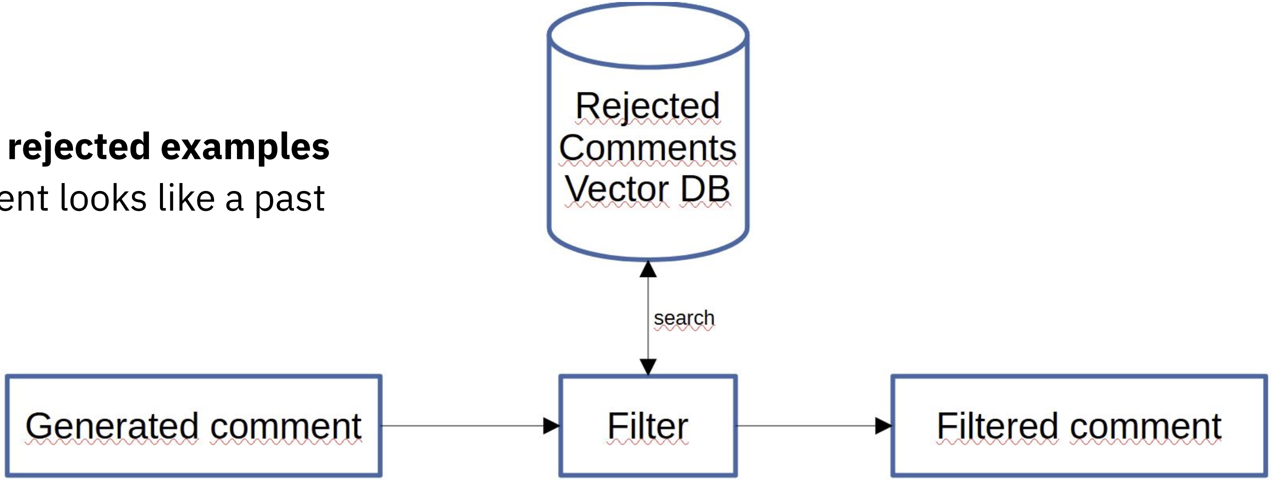
- Can be used to build **highly valuable examples** (issues missed during review)



Reduce “trivial” and unwanted

- Allow **customization by type**
    - E.g. some devs want to see tips, some don’t

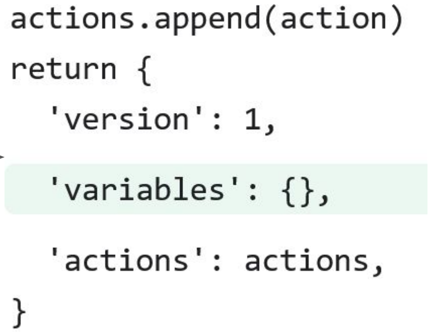
- More filtering through **past rejected examples**
    - If a generated comment looks like a past



Autofixing comments

- Both human comments and LLM comments

`variables` is actually required (see https://hg.mozilla.org/mozilla-central/rev/2c3c21948ede) but can be an empty dict.



```
actions.append(action)
return {
  'version': 1,
  'variables': {},
  'actions': actions,
}
```

Development time vs review time

- Many developers want to run the tool **before submitting for review**

- **IDE integration**

- Detect issues **ahead of time**, avoid wasting reviewers time

Agenda

01 A trip down prompt engineering lane

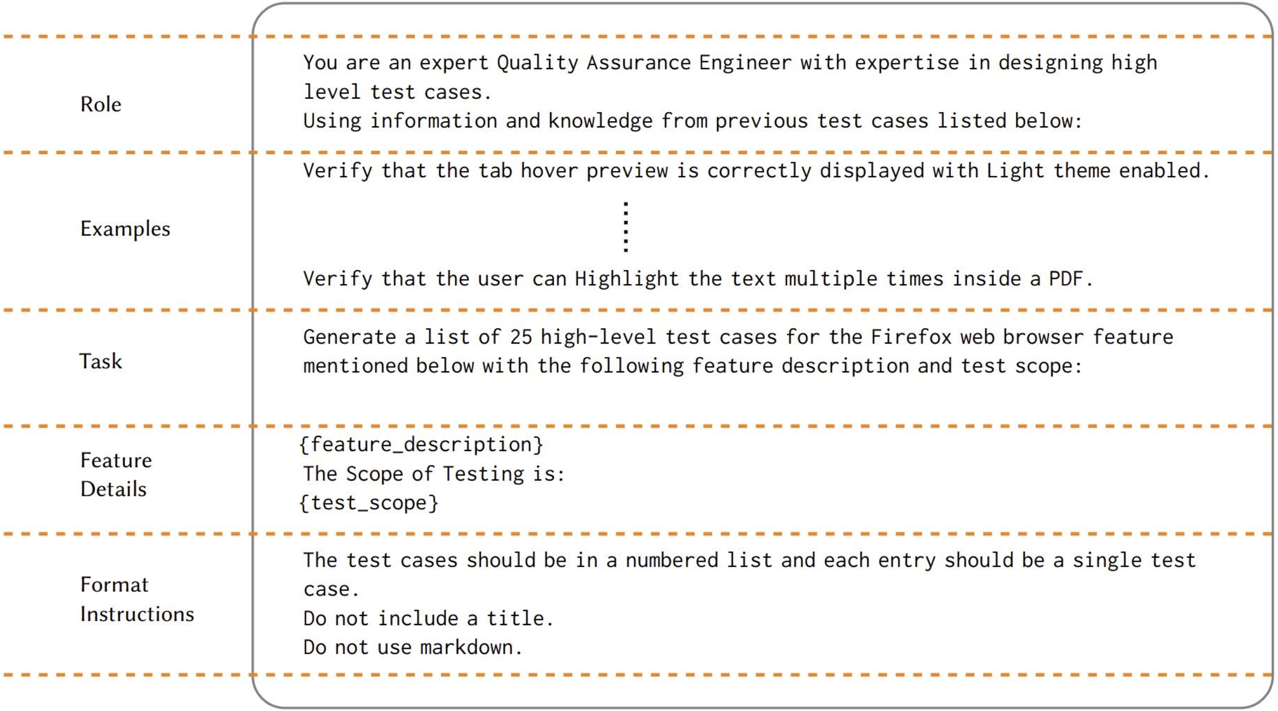
02 What it looks like

03 Do people like it?

04 Next steps

→ 05 What else?

QA test case generation

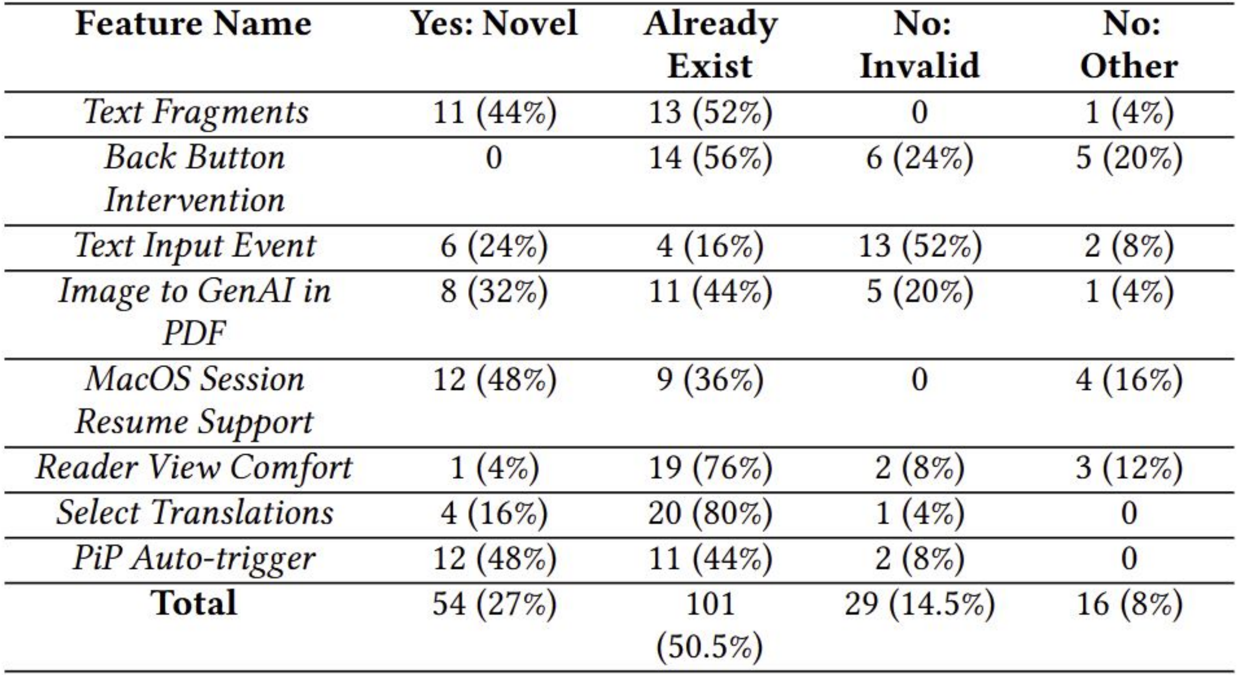


Role	You are an expert Quality Assurance Engineer with expertise in designing high level test cases. Using information and knowledge from previous test cases listed below:
Examples	Verify that the tab hover preview is correctly displayed with Light theme enabled. ⋮ Verify that the user can Highlight the text multiple times inside a PDF.
Task	Generate a list of 25 high-level test cases for the Firefox web browser feature mentioned below with the following feature description and test scope:

Feature Details	{feature_description}
	The Scope of Testing is:
	{test_scope}

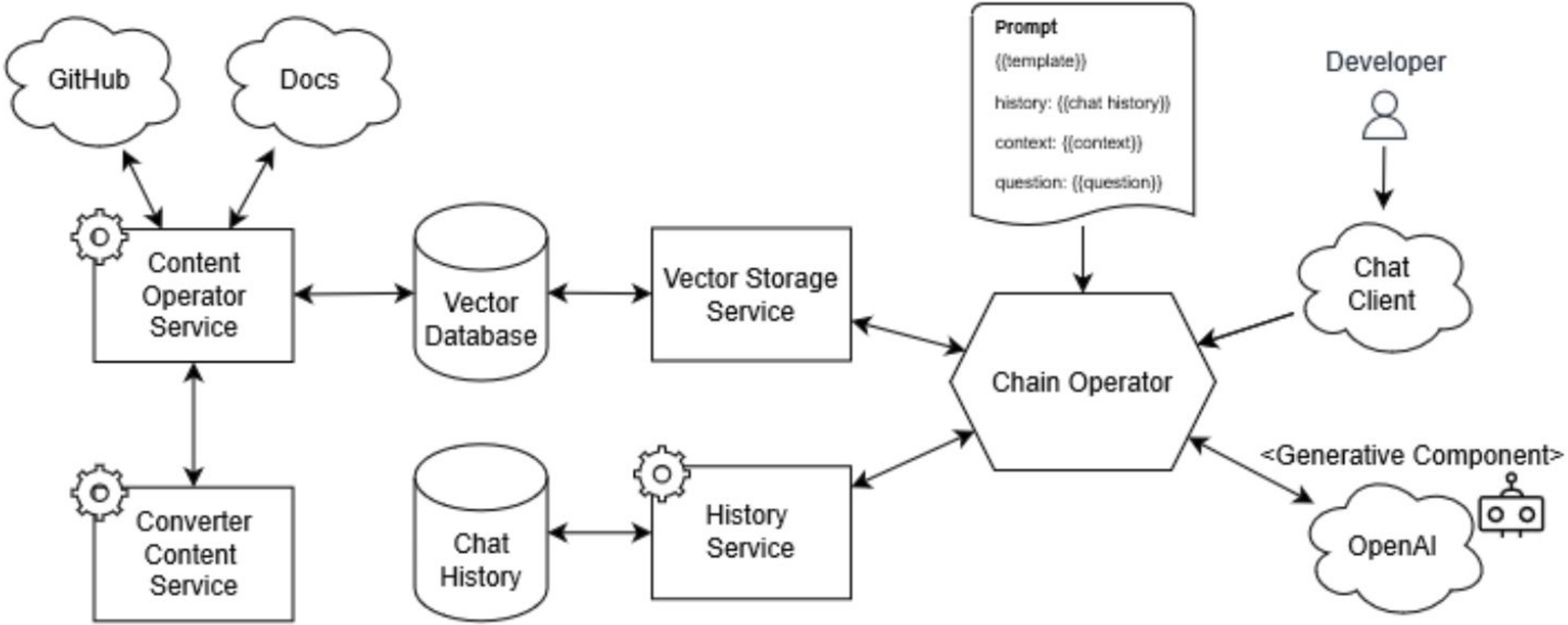
Format Instructions	The test cases should be in a numbered list and each entry should be a single test case. Do not include a title. Do not use markdown.

QA test case generation



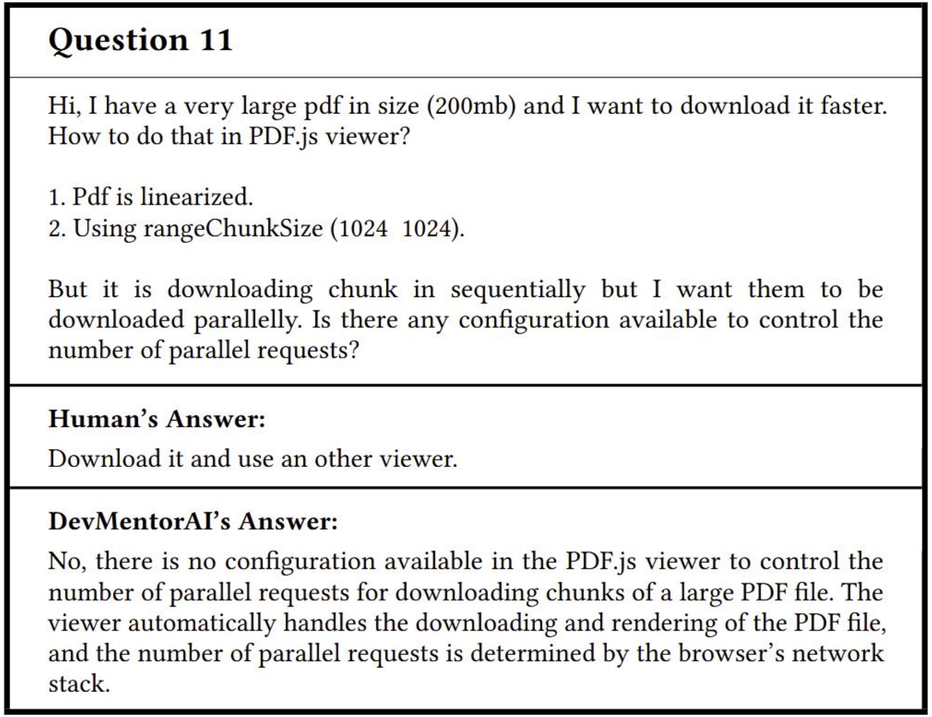
Feature Name	Yes: Novel	Already Exist	No: Invalid	No: Other
Text Fragments	11 (44%)	13 (52%)	0	1 (4%)
Back Button Intervention	0	14 (56%)	6 (24%)	5 (20%)
Text Input Event	6 (24%)	4 (16%)	13 (52%)	2 (8%)
Image to GenAI in PDF	8 (32%)	11 (44%)	5 (20%)	1 (4%)
MacOS Session Resume Support	12 (48%)	9 (36%)	0	4 (16%)
Reader View Comfort	1 (4%)	19 (76%)	2 (8%)	3 (12%)
Select Translations	4 (16%)	20 (80%)	1 (4%)	0
PiP Auto-trigger	12 (48%)	11 (44%)	2 (8%)	0
Total	54 (27%)	101 (50.5%)	29 (14.5%)	16 (8%)

Smart documentation search



Smart documentation search

✓



Question 11

Hi, I have a very large pdf in size (200mb) and I want to download it faster. How to do that in PDF.js viewer?

1. Pdf is linearized.
2. Using rangeChunkSize (1024 1024).

But it is downloading chunk in sequentially but I want them to be downloaded parallelly. Is there any configuration available to control the number of parallel requests?

Human's Answer:

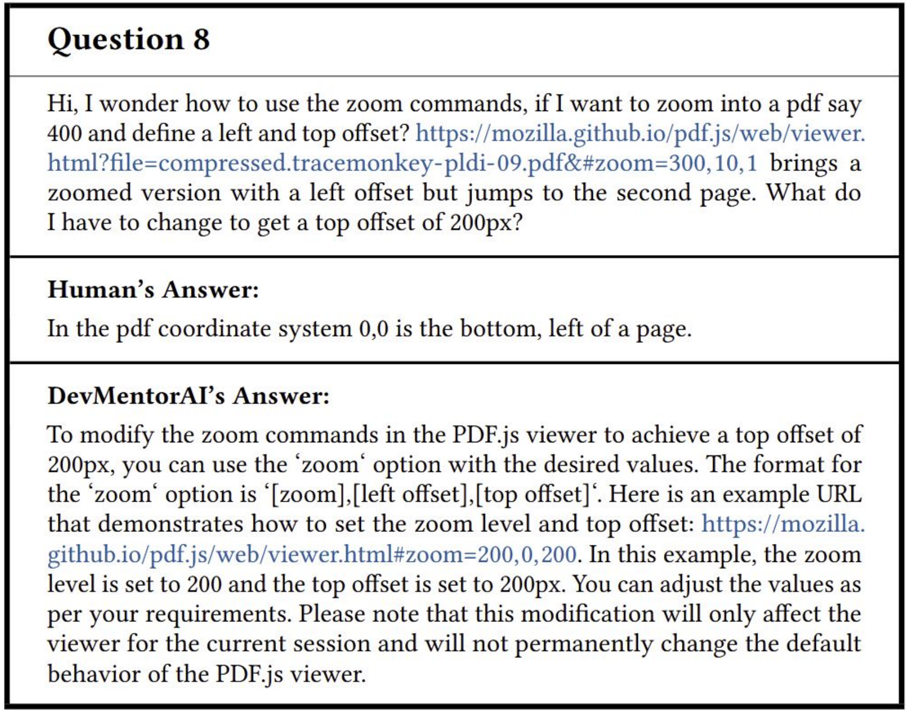
Download it and use an other viewer.

DevMentorAI's Answer:

No, there is no configuration available in the PDF.js viewer to control the number of parallel requests for downloading chunks of a large PDF file. The viewer automatically handles the downloading and rendering of the PDF file, and the number of parallel requests is determined by the browser's network stack.

Smart documentation search

(bad docs)



Question 8

Hi, I wonder how to use the zoom commands, if I want to zoom into a pdf say 400 and define a left and top offset? https://mozilla.github.io/pdf.js/web/viewer.html?file=compressed.tracemonkey-pldi-09.pdf&#zoom=300,10,1 brings a zoomed version with a left offset but jumps to the second page. What do I have to change to get a top offset of 200px?

Human's Answer:

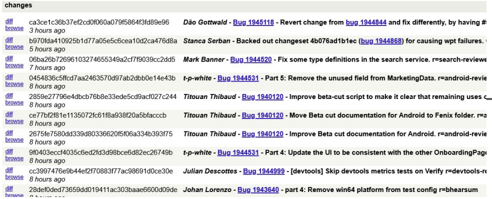
In the pdf coordinate system 0,0 is the bottom, left of a page.

DevMentorAI's Answer:

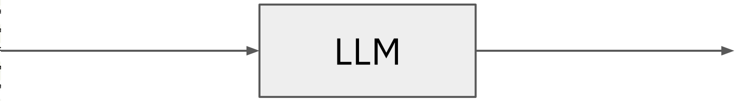
To modify the zoom commands in the PDF.js viewer to achieve a top offset of 200px, you can use the ‘zoom‘ option with the desired values. The format for the ‘zoom‘ option is ‘[zoom],[left offset],[top offset]‘. Here is an example URL that demonstrates how to set the zoom level and top offset: https://mozilla.github.io/pdf.js/web/viewer.html#zoom=200,0,200. In this example, the zoom level is set to 200 and the top offset is set to 200px. You can adjust the values as per your requirements. Please note that this modification will only affect the viewer for the current session and will not permanently change the default behavior of the PDF.js viewer.

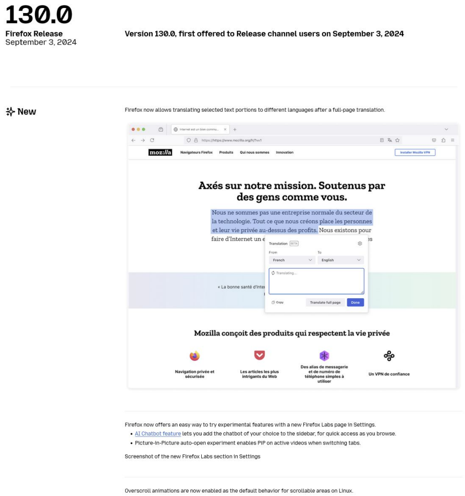
Identifying important changes

- **46’000 commits last year**, how do we build release notes?



```
changes
diff    ca3ce1c36b37ef2cd0f060a079f5864f3fd89e96   Dão Gottwald - Bug 1945118 - Revert change from bug 1944844 and fix differently, by having #
browse  3 hours ago
diff    b970fda410925b1d77a05e5c6cea10d2ca476d8a   Stanca Serban - Backed out changeset 4b076ad1b1ec (bug 1944868) for causing wpt failures.
browse  5 hours ago
diff    06ba26b72696103274655349a2cf7f9039cc2dd5   Mark Banner - Bug 1944520 - Fix some type definitions in the search service. r=search-reviewe
browse  7 hours ago
diff    0454836c5ffcd7aa2463570d97ab2dbb0e14e43b   t-p-white - Bug 1944531 - Part 5: Remove the unused field from MarketingData. r=android-revie
browse  8 hours ago
diff    2859e27796e4dbcb76b8e33ede5cd9acf027c244   Titouan Thibaud - Bug 1940120 - Improve beta-cut script to make it clear that remaining uses c
browse  8 hours ago
diff    ce77bf2f81e1135072fc61f8a938f20a5bfacccb   Titouan Thibaud - Bug 1940120 - Move Beta cut documentation for Android to Fenix folder. r=a
browse  8 hours ago
diff    2675fe7580dd339d80336620f5f06a334b393f75   Titouan Thibaud - Bug 1940120 - Improve Beta cut documentation for Android. r=android-revie
browse  8 hours ago
diff    9f0403eccf4035c6ed2fd3d98bce6d82ec26749b   t-p-white - Bug 1944531 - Part 4: Update the UI to be consistent with the other OnboardingPage
browse  8 hours ago
diff    cc3997476e9b44ef2f70883f77ac98691d0ce30e   Julian Descottes - Bug 1944999 - [devtools] Skip devtools metrics tests on Verify r=devtools-re
browse  8 hours ago
diff    28def0ded73659dd019411ac303baae6600d09de   Johan Lorenzo - Bug 1943640 - part 4: Remove win64 platform from test config r=bhearsum
browse
```





130.0

Firefox Release
September 3, 2024

Version 130.0, first offered to Release channel users on September 3, 2024

New

Firefox now allows translating selected text portions to different languages after a full-page translation.

Axés sur notre mission. Soutenus par des gens comme vous.

Nous ne sommes pas une entreprise normale du secteur de la technologie. Tout ce que nous créons place les personnes et leur vie privée au-dessus des profits. Nous existons pour faire d’Internet un e

« La bonne santé d’Inter

Mozilla conçoit des produits qui respectent la vie privée

Internet est un bien commu...
https://https://www.mozilla.org/fr/?v=1
moz://a
Navigateurs Firefox
Produits
Qui nous sommes
Innovation
Installer Mozilla VPN
Translation BETA
From
French
To
English
Translating...
Copy
Translate full page
Done
Navigation privée et sécurisée
Les articles les plus intrigants du Web
Des alias de messagerie et de numéro de téléphone simples à utiliser
Un VPN de confiance

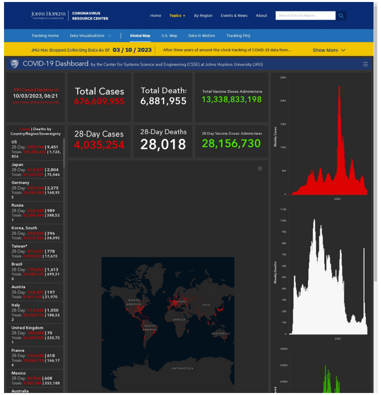
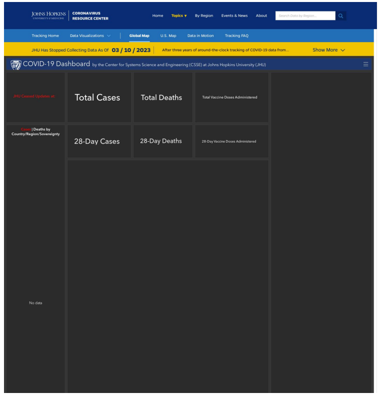
Firefox now offers an easy way to try experimental features with a new Firefox Labs page in Settings.

- AI Chatbot feature lets you add the chatbot of your choice to the sidebar, for quick access as you browse.
- Picture-in-Picture auto-open experiment enables PiP on active videos when switching tabs.

Screenshot of the new Firefox Labs section in Settings

Overscroll animations are now enabled as the default behavior for scrollable areas on Linux.

WebCompat issue detection

Chrome Screenshot	Firefox Screenshot
	

JOHNS HOPKINS
UNIVERSITY & MEDICINE
CORONAVIRUS RESOURCE CENTER
Home
Topics
By Region
Events & News
About
Search Data by Region...
Tracking Home
Data Visualizations
Global Map
U.S. Map
Data in Motion
Tracking FAQ
JHU Has Stopped Collecting Data As Of 03 / 10 / 2023
After three years of around-the-clock tracking of COVID-19 data from...
Show More
COVID-19 Dashboard by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU)
JHU Ceased Updates at:
Total Cases
Total Deaths
Total Vaccine Doses Administered
Cases | Deaths by Country/Region/Sovereignty
28-Day Cases
28-Day Deaths
28-Day Vaccine Doses Administered
No data

S2 (Serious). The missing data tables and charts in Firefox severely impair the usability of the dashboard. A significant portion of the data is unavailable to users using Firefox, rendering a critical part of the website non-functional

Autofixing the world

- Extend autofixing beyond review comments to additional cases, e.g.:
    - Identify the cause of a **crash** from a stack trace and suggest a fix
    - Analyze **build failure** logs for a patch to determine and resolve the issue

- **Feedback cycle** to help the LLM (e.g. rebuild, share new error log)

End of the Journey. Interested?

- Want to experiment with our stuff and see where it could go?
- Want to expand support for our tools to other review systems and projects?
- Want to adopt our tools in your projects?
- **Come talk to us in our devroom!**

Thank you