

Ecosystem Infra bi-weekly

Schedule: <https://bit.ly/ecosystem-infra-talks>

2019-05-28

Attending: Simon Pieters (presenting), Philip, Robert, Luke Z, Sam, Kyle, Mike

Presentation: [-webkit-appearance](#)

2019-05-07

Attending: Luke B, Luke Z, Robert, Kyle, Sam, Simon, Mike

No presenter today, so we'll talk about things at random.

Luke B: There was a Microsoft blog post with [link to WebView2](#).

Robert: 2nd gen WebView for win32 applications powered by Chromium Edge. Developers can choose between system evergreen engine, or bundle a specific version. Seems like Electron a bit.

Philip: Is the system one updated less often?

Robert: No, either evergreen or bring your own.

Sam: Sounds like a play to make it possible to ship Electron things without bundling all of Chromium.

Robert: That is the upside if some chose to use the evergreen.

Philip: Sounds like more stability when building using web technology?

Luke B: Would be better if the web was considered stable enough, but mixed feelings.

Robert/Sam: Could add things that aren't available to the web.

Robert: Another question is with Edgium is it'll be stable soon, so what should we do?

Philip: Heard from John/Mustapha that getting it running on Windows Server was tricky because of something window manager.

Sam: Heard goal was autumn update Windows release.

Luke: Would the Azure Pipelines carry over to Edgium when that comes around?

Philip: Most likely not, if we do nothing (and current setup works) it'll just keep running EdgeHTML.

Luke: If Edge and Chrome on Windows are same engine Window, value of running both?

Sam: We've talked about running other Chromium-based browsers. Is Edge more important than Samsung Internet? Probably not?

Sam: Should we drop EdgeHTML from home page?

Luke: And should we drop Edgium? Our consumers are browser engineers, so distinction between Edge and Chrome isn't big enough to help inform priorities.

Philip: Best case perhaps that it's not on the home page but MS still finds it valuable to contribute results.

Luke: Don't anticipate large differences between Linux and Windows, but comparing them could be valuable. Maybe we should ask Mozilla, having two Chromiums might skew results.

Philip: For Firefox-only failures?

Robert: We have metrics where we have 4 browsers which might not make as much sense.

Mike: (1) what's technically useful and (2) politics and health. There's also Servo, instead of dropping from 4 to 3 could put in Servo. wpt.fyi isn't trying to be caniuse.com and not serving web developers.

Philip: Poked jdm a while ago to see if they'd be keen to send results, but none yet. Question is if we should work on it if Mozilla doesn't.

Robert: It's not used in any real product right?

Sam: Still a research project, more bits going into Firefox.

Luke: We could make the argument that regular Servo results will improve health of running infra, but that'd be true for Windows as well. In that past we've not prioritize because of relevance of Servo.

Sam: There's probably more value in Android testing.

2019-03-19

Attending: Robert, Luke Z, Kyle, Philip, Simon, Mike

Robert presenting on [wpt.fyi reftest screenshots](#).

Robert: A lot of discussion at TPAC about reftest screenshots. Now it's here!

2019-03-05

Attending: Kyle, Philip, Luke Z, Luke B, Robert, Sam, Mike

Kyle presenting [WPT WebDriver Test Runner & Triage Meta-data](#)

Discussion:

- Philip: Can we search for things that don't have links?
- Luke: Yes that's important for triaging new failures, !link:bugzilla would do that.
- Philip: Tradeoff between linking to file names (.any.js) vs. test names (.window.html)?
- Luke: Since people are triaging failures which are in terms of tests it makes most sense to express metadata in the same domain.
- Mike: There's a lot of variability in subtest names and existence.

- Luke: Yeah. Right now we can only search by test name and not subtest name, so want to match that.
- Mike: How did you decide on the MVP?
- Luke: Three different verticals, and also James' wptdash was input to it, and Sam's input. But could add and remove things, not heavily researched.
- Mike: Have you thought about linting, so that there's no extra work? Risk of things going stale. When building the index it's too late to find the problems.
- Luke: Kyle touched on that, we'd like to stick to tip of tree. I'd like to have a daily pipelines that finds orphaned results or stale results. MVP is ability to add data.
- Philip: For adding data, do we need to provide some scripts?
- Luke: Long term I'd like something like the query builder, a graphical builder. In terms of MVP each vendor could crawl their bug database and upload links.

2019-02-12

Attending: Philip, Joe Medley, Florian Scholz, Simon, Sam Sneddon, Mike, Kyle, Luke Z, Robert

Philip presenting about [Assisting BCD updates with automation](#)

Joe: this isn't a trivial problem that can be totally solved; sizable gradual improvements are appreciated.

Mike: how do you define "good enough" support of a feature?

Philip: this is why we can't go directly from WPT to BCD

Joe: it's a non question: the information is best-effort; the developers need to make the judgement call

Mike: but since the presentation is a red/green yes/no, aren't we making the call on devs' behalf?

Joe: another benefit of having the earliest version number supporting a feature: another signal for web developers (a feature that's been supported longer is perhaps safer to use)
(more discussion about subfeatures vs edge cases)

Philip: human judgement is the advantage of BCD

Sam: since web IDL, lots of things have been moved to the prototype. In addition to web IDL, we need to know how to construct an instance of an interface.

Prefixes are another special case that's worth handling.

Simon: a percentage of WPT passing typically means nothing to web developers

Mike: if we lower our standards ("good enough") for web developers, are we lowering standards for the ecosystem?

Florian: but if we require say 100% WPT passing then we can't recommend anything

Mike: perhaps more annotation in WPT to help people to make the judgement call

Florian: remember it's not a judgement call by Mozilla; BCD is open-source and many contributions come from the community already

2019-01-29

Attending: Luke Z, Robert, Kyle, Mike, Philip

Luke Z presenting [WPT in Chromium CI](#)

2019-01-15

Attending: Mark, Robert, Lukasz, Philip, Kyle, Boaz, Luke, Sam

Around the room: new year's resolution for the web platform:

- Philip: boring and metric-y, but reduce number of browser-specific failures in wpt
- Boaz: something about Edge using Chromium, collaboration, governance
- Lukasz: get some agreement on a 2-way protocol for driving browsers
- Robert: People will start writing tests that work cross-browser by default. Maybe by standardizing protocol, or something else.
- Kyle: I support!
- Mark: I think wasm will ship useful types.
- Luke: establish a story for testing Edge in the world of Edgium.
- Sam: be more interoperable!

Mark presenting on [Structured Queries on wpt.fyi](#)

2018-12-18

Attending: Philip, Sam, Mike, Luke, Kyle, Robert, Lukasz, Mark

Luke presenting on [wpt.fyi status checks](#).

2018-12-11

Attending: Marie, Mike, Luke, Mark, John Chen, Philip, Sam, Boaz, Robert

Mike presenting [WPT on CDP](#)

2018-11-27

Attending: Robert, Mark, Luke B, Luke Z, Sam, Boaz, Mike, James, Philip, Kyle

Taking notes: Philip

Robert presenting [Blink survey re: LayoutTests & WPT and wpt-puppeteer](#).

2018-11-06

Attending: Tab, Philip, Sam, Mike, Luke, Lukasz, Mark, Robert, Kyle

Tab presenting on testing CSS.

Tab: At [CSSWG/WPT joint meeting](#), discussion was similar to last year, as we're a small group with a lot more connection to web developers than browser vendors. A compromise that seems reasonable and that I've agreed to is to at least have *one* test, which implementers would notice.

Tab: A problem is PR review. Idea in the meeting was to [automatically approve tests from CSS editors](#), on the assumption that as soon as an implementer looks at it, they'll notice it's no good.

Tab: With those changes (block on tests, at least one test per PR, bypassing reviewing) we should be able to get CSS into a much better shape.

Philip: Tradeoff for dedicated QA vs. having editors be on the hook?

Tab: Options are to decrease editor output, increase editor pool, or hiring QA people. The last is the most reasonable in that you can do it with a shorter time horizon. Can tackle the backlog in this way.

Philip: If we burn down backlog in some areas (Flexbox and Grid), would ongoing maintenance also require ongoing extra testing resources?

Tab: Think we should be able to keep up.

Luke: Who is expected to take on the extra cost of testing resources?

Tab: Don't know.

Luke: Seemed like in the meeting, nobody put up their hand to hire these resources. There was some effort there that didn't work? I think someone mentioned that.

Luke: You said skill required to write test isn't as high to write specs. Do we have evidence for that? It seems to me that quality of test is proportional to the understanding of the thing you're writing a test for.

Tab: Yes. But writing good specs is a different skill set, technical writing which not everyone is good at even if they're a good test writer.

Luke: Does an analogy between designer and web site implementer work here? Some test frameworks are a bit abstracted from the implementation details. Would something like that help smooth the gap, or do you need domain knowledge?

Tab: In some cases domain knowledge is unavoidable.

Sam: Often you have to follow 40 references and understand things like initial containing block.

Tab: Yes, but it's still less ramp up work than becoming a long term editor.

Mark: You mentioned reluctance to lower the pace of the WG. Do you think there's a cost to the ecosystem from doing less new work, or more like editors not wanting to write tests?

Tab: It's actually not just new work, there's also a lot of maintenance work, where reaction time matters. The slower we fix things the worse it gets.

Mark: So cost to ecosystem is that the WG can get overrun with fix work?

Tab: Yeah, and we're already in that case.

Mike: This seems like the area where automatic review would be the most preferable, despite my comments about that on the issue. What I think we'd lose out on isn't correctness of tests, but correctness of the specification. Best sort of testing review allows you to review your own assumptions. Validating your own assumptions loses some of that. OK with this idea for backfilling work, rather than for new work.

Tab: I agree, leaving self-review for backfilling makes sense. If I'm writing a comprehensive new test suite, I'd like someone else to review it.

Sam: Do think it'll be easier to get someone to pay for testing work than funding long term editors?

Tab: If nothing else it's easier logistically, non-open-ended projects to "sell."

Philip: Would practice around pushing to master directly have to change?

Tab: Not for auto-approved test PRs, but if someone else is going to review that makes sense.

2018-10-16

Present: Philip, Luke, Sam, Kyle Ju, James Graham, Mike P, Boaz S,

Presenting: Philip J, [Expanding wpt's CI platform coverage](#)

Discussion:

- Foolip: my recommendation is to move safari tp onto. I think it would be good to be on microsoft's CI in order to get them to upgrade to windows 10/edge
- Boaz: I would double check that.
- James:
 - historically the issue with windows server is a licensing issue
 - I think this is a good plan
 - We should be trying to move our linux runners off of travis, (we can't write http2 tests on travis)
 - Looks like azure pipelines is a better product
 - It seems like the main thing tying us to any choice is the os/price matrix
 - Foolip: the possibility of getting windows 10 is maybe more important
- Boaz: how stable is wpt on osx and windows 10
 - Philip: stable

- Luke: Are you able to scrape azure?
 - Philip: sort of
- James:
 - We want azure to support conditional runs, and longer build times
 - If we're thinking of using this for macosx, then we should definitely use this for windows
- Foolip:
 - I also would like to.. [clipped]
- James
 - I know macos has issues with picking up fonts
 - On gecko we chuck windows into 12 chucks and 1/12 of wpt takes 17 minutes
- Boaz
 - I expect we'll want to invest in making wpt run faster.

2018-10-02

Present: Mark, Kyle, Robert, Boaz, Sam, Philip (taking notes)

Discussion topic: Chose one web platform feature to remove to improve web page performance, what would it be? If you break the web please suggest a replacement!

- Mark: I'd replace the CSS cascade with a less leaky abstraction
- Robert: all synchronous APIs. Sync XHR.
 - Philip: <https://github.com/whatwg/xhr/issues/20>
- Philip: I was going to say sync xhr. Floats in CSS? (Sam says no.)
- Sam: The default layout mode, because it's $O(n^2)$ in number of rows.
- Boaz: Images? Event binding model? GC? Ads?
- Kyle: Lazy loading of JS perhaps.

Mark presenting today on interop data from confluence

Mark: what patterns would be interesting to look for in this spreadsheet, and why?

Boaz: where's this data from?

Mark: <http://web-confluence.appspot.com/#/> (existence of APIs)

Philip: APIs that start out at 0 or 1 and reach 3 but never 4 would be great, these are cases where a single vendor can be prodded to give us full interop.

Philip: Can we subset this data somehow, with MDN or something else? Perhaps we could see the APIs for some feature or some spec behaving quite differently in terms of interop evolution over time.

Mark: Linkage has been coming up a lot recently.

2018-09-18

Mike presenting today on The Long Road to Async Cleanup

Discussion: Which is your favorite testing framework and why?

- Luke: a long time ago I used a semantic framework called [specflow](#). test files were “given a premise, when some action, then some outcome”. Strings would map to code to assert it, but the tests could be written by PMs. “If I don’t have an account, I’ll be redirected.” Similar to rspec.
- Kyle: Used to do iOS development, used Jenkins. First time in this meeting, transferred into the team recently.
- Robert: [Mocha](#). I haven’t used it much, but when I was an intern on Taskcluster we wrote most JS tests in mocha. Supports both TDD or behavior-driven design. Maybe too much flexibility, people were arguing about flavors. Probably good for small team.
- Mike: That’s my favorite too. Also like pytest.
- Boaz: Don’t have a favorite, last I used QUnit. Since then have just used whatever has been set up. I like the frameworks that make my coworkers happy.
- Philip: Mocha. I like that it separates the assertion library from the test structure library. Writing async tests in mocha is nice. I like the output, npm test with mocha looks nice.
- Mark: skipped.
- Mike: I was pretty sure nobody was going to say testharness.js. Today want to talk about how we made testharness.js just a little bit nicer.

2018-09-04

Present: Luke, Mark, Robert, Mike, Sam

Round table: workflow tip!

- Mark: hard limit on number of properties on a web component! Somehow?
- Robert: gofmt isn’t stable! Apparently we need to vendor a fixed version of gofmt. (This would have been a “tip” last week, but now it’s really a complaint.)
- Mike: set cookies:
<https://github.com/web-platform-tests/results-collection/issues/599#issuecomment-418219542> (cookies set w/ and w/o domains are in fact separate)
- ~~Sam: browser versions?~~
- Luke: VS code: beautify
<https://marketplace.visualstudio.com/items?itemName=HookyQR.beautify>

Topic: [\(MVP\) wpt.fyi v1.0?](#)

Mike: UI for selecting runs to diff

Luke: On a high level, in addition to viewing & comparing results, anything else? Bikeshed?

2018-08-28

Present: Robert, Philip, Luke, Sam, Boaz

Topic today: Use of wpt in blink

Round table: if our team had just one success metric, what would it be?

- Boaz: Number of browser engines using wpt to prioritize their work.
- Sam: percentage of modified tests being written which are wpt in all engines?
- Robert: Same as Sam :)
- Sam: Percentage of tests passing in all 4 browsers.
- Boaz: quality of specs: test coverage + implementation coverage (we have some from Chromium and Firefox)
- Philip: Web developers are spending less time struggling with interop (survey based).
- Luke: Number of web properties using polyfills. Absolute number or proportion of “large” web properties. Hypothesis being that polyfills are smoothing over interop problems.

Ideas from presentation (how to get blinkers to use WPT)

- Philip: Put a readme.wpt in the target test directories
- Robert: webxr/scroll-snap are new since May, but for some reason, those teams started with non wpt layout tests. Why is this? They didn't know they could put experimental tests in WPT?
- Boaz: ask teams with older test suites to add WPT based KRs.
 - Robert: webxr has this
- Mike: Ask these teams to use testharness.js
- Mike: potentially there are cultural differences in test harness issues?
 - Philip: people find jstests more convenient since you only have to print the value. Also, the expectation args order is reversed.

Post-meeting update: [LayoutTests/fast/scroll-snap/](#) has only 3 tests, <https://wpt.fyi/results/css/css-scroll-snap> has the tests that could be upstreamed.

2018-08-21

Present: Mike, Sam, Philip, Tab (presenting)

Topic today is wpt.fyi results API, to see if there's anything needed to make the Bikeshed <wpt> feature work. What's missing now is the ability to link to a subtest directly as a single test might be testing many different things. Suggest perhaps the first 10 chars of a hash of the test name.

Philip: Is it a problem that this might change a lot?

Tab: It would be a bit of a problem.

Robert: Another benefit of hashing is that some subtest names have weird characters.

Mike: Distinction between test and subtest is a technical one currently. Can't procedurally generate tests, but can generate subtests. We have an opportunity to introduce a semantic difference. Requiring the tests to be organized similar to the spec would sidestep this problem. Also relates to the "is this even tested yet" problem.

Tab: That's valid, might want to be able to match all subtests, or all tests in a directory.

Mike: Like supporting that pattern of test authoring.

Philip: In some context the idea of having nested subtests for testharness.js, similar to mocha and other JS testing frameworks.

Tab: Back to the beginning, can we have subtest links for wpt.fyi? File a bug?

Philip: Yes, file a bug. Think it'd be the /api/results API modifying the results we have in GCS before returning them to the client.

Mike: Big idea: could we have the wpt directory structure match spec algorithm IDs, so that Bikeshed could get this structure from wpt without any manual linking?

Tab: I'd really like that, how automatic it could be.

Mike: Would it be a problem that the same test can only appear in one place?

Tab: We have some tests now that belong in two places.

2018-08-14

Present: Robert, Jon, Philip, Boaz, Sam, Mike

Round-table: Anything exciting/frustrating in the past two weeks?

- Jon: movement on <https://github.com/Honry/sensors/pull/1> is exciting!
- Boaz: Simon Pieters is back and working on HTML fieldset interop. I have been reaching out to implementers to find engineers. Difficult to get synchronous attention!
- Mike: A lot of frustrating things. Sauce Labs released a new thing which I updated, and verified the checksum. But the checksum changed last week. So they changed the checksum on the website. Told them it's pointless :)
- Philip: Excited that reffy is publishing CSS syntax as JSON now, so we can generate "cssharness" tests.
- Sam: Annoyed with ChromeDriver bugs :) Have only needed to work around ChromeDriver issues, not EdgeDriver or SafariDriver.
- Robert: Picked this issue because I've had a lot of frustration. Have been trying to do a prod release for wpt.fyi for two weeks. Have had real bugs, Travis network issues, gcloud SDK issues...

Presenter: Robert Ma

Presentation: [A Case Study on Hard-to-test APIs](#)

2018-07-31

Present: Robert, Jon, Luke, Philip

Round-table: what are some arguments for having teams use WPT / wpt.fyi?

- Luke: Differences between master of chromium (content_shell) and Chrome stable, so comparing with other browser and other browser in a stable version.
- Philip: The small amount of extra work due to being a shared test suite means the tests can be run in other browsers, you can see the results, and other vendors can implement faster, so your features becomes available to web devs sooner.
- Robert: Using WPT is a very convenient way to share the tests, leveraging 2-way sync. But also seen teams that don't care about wpt.fyi per se, because there's some special setup required, so it only works in everyone's own CI, but not in wpt.fyi.
 - Jon: Don't we have plans to run ever-more configurations?
 - Robert: Yeah. But things like WebUSB... will rely on some progress on WebDriver and testdriver.js.
- Jon: Collect all the good reasons and put it in a doc. And figure out which Chrome teams use it, and not.

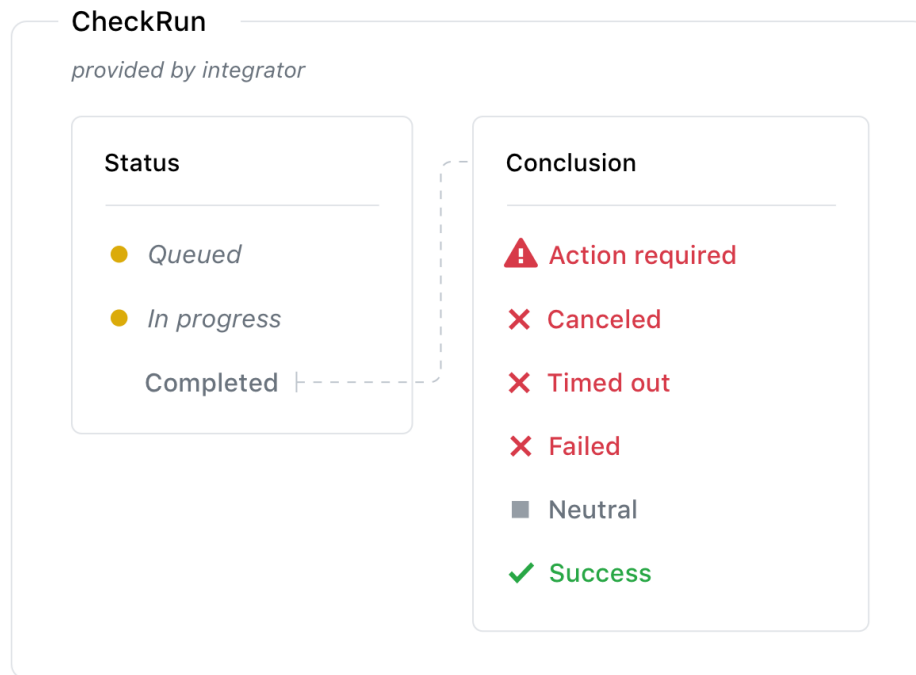
Jon presenting on WebDriver Extensions

2018-07-24

Present: Philip, Jon, Boaz, Mike, Robert, Luke, Boaz, Brendan, Mike

Philip on the GitHub checks API. "Slides":

- Want to do this for my "[GitHub status check for spec PRs to link to tests and browser bugs for failures](#)" KR.
- <https://developer.github.com/v3/checks/>



- If a GitHub app has write permission for checks and is configured to get webhooks for requested check, checks will be shown as [pending on PRs](#). There's also a re-run button. (With Travis retrying requires you do go to the Travis page and authenticate there.)
- The GitHub app then needs to create the check and update when it finishes.
- The interesting part is the check output, which can be much more rich than pass/fail:
 - A title, but also a summary and details including markdown. Which means one can link to things on wpt.fyi and that sort of thing.
 - Annotations on specific lines of code. Most obviously, could be used to make lint errors nicer.
 - Images with captions. Not immediately useful? Memes!
- There are also up to three actions that turn into buttons in the UI, which seems like it will then turn back to the GitHub app to perform the action. So probably "run autopep8 --in-place to fix the problem you found" could be one such action.

Round table question: given this new checks API, what are some new things we could do?

- Jon: Sounds like in the spec repo case it could be really spammy?
- Boaz: One day, when wpt is in better shape, prompt people to go do the next step. So a button that files bugs on browsers.
- Brendan: On wpt, only affected tests are run again. But if you wanted to run all tests, have that as a simple button.
- Robert: A bit more short term, a tighter integration between Travis and GitHub, if we use the checks version of Travis, then we could probably make the checks easier to use. For the stability check we're already printing out Markdown in the logs. So put that into the checks API.

- Luke: (Philip: don't need a per-PR results page on wpt.fyi with this?) There's probably still value in navigating to wpt.fyi for the full results. But can see the summary on GitHub.
- Mike: On w3c-test.org, could perhaps make the deployment of /submissions/ by on request instead of doing it for every PR. Because spinning up a Docker image for every PR might not be a great idea. Luke: on wpt.fyi we deploy PRs automatically, but no cleanup method. A button to clean it up would be great.
- Robert: Also solves some of the spam problem.
- Philip: Will go poke Travis about whether we can get the Checks version of Travis sooner rather than later. (After the meeting: found <https://blog.travis-ci.com/2018-05-07-announcing-support-for-github-checks-api-on-travis-ci-com> which says it's for travis-ci.com, and we're on travis-ci.org for now.)

2018-07-17

Back after two weeks of no meetings. Mark presenting today.

Present: Mark, Luke, Robert, Brendan, Mike, Philip, Jon

Around the room: Give example of a specific time when trying to find results about a specific something, where it didn't deliver.

- Mark: Want to search by API names and features.
- Luke: No easy way to group similar tests across all APIs. Hard to find a similar test to use as an example. So searching by contents of files.
- Robert: Would like search to go deeper into directories, and popping out again if you remove parts of your search.
- Jon: Historical data, ability to see how it has changed over time. Key changes with dates. Both changes in WPT and changes in browsers, ideally.
- Philip: Full diffs of runs that show just the changes, hiding what hasn't changed, kind of like a textual diff.
- Brendan: Searching for tests.
- Mike: Like Luke, don't have a way to find tests for features spread over specs, like Shadow DOM or Service Workers.

Mark presenting new features for the dashboard:

<https://chart-ux-dot-wptdashboard-staging.appspot.com/results>

2018-06-26

Luke presenting today.

Prompt: Success stories about using WPT?

- Boaz: authored selectionChange API test: <https://github.com/web-platform-tests/wpt/pull/11491> which was the first one for the event

in WPT and shook out a lot of interop bugs and todos
(<https://github.com/web-platform-tests/wpt/issues/11493>)

- Sam: wptserve & py3 WIP!
- Robert: I got UC browser test results, it took them a week
 - Boaz: I would love to speed that up
 - Mike: can you run android on a PC (X86)?
 - Sam & Robert: Yes, but probably more interesting to run on ARM (e.g. floating points, etc.) [gsnedders adds: (x86+SSE2 or x86_64) v. ARM shouldn't have fp differences]
- Luke: I successfully was able to use our admin to upload results!
- Mark: about <https://github.com/web-platform-tests/wpt.fyi/issues/62> (counting harness status & test status separately)

Presentation

Boaz: thinking about web developers: more powerful data APIs (summary/aggregation, ability to query a feature)

Mike: full results in API (Robert: yes we just released that in prod! E.g.
<https://staging.wpt.fyi/api/runs?product=chrome>)

Mark: documentation!!

Sam: more regular Android X86 runs and less often ARM runs

2018-06-19

Robert presenting today.

Prompt: "What would you like to improve in your daily WPT workflow?" Bonus points for how.

- Robert: Very unsatisfied with notifications from GitHub. Hard to filter emails. No desktop notifications. Have very complicated filter rules.
- Mark: Editor support? Catching breaking changes across repos in the web-platform-tests org. Some way of noticing those breaking changes in advance.
- Simon: [Auto-merging when status checks are green](#). Luke: can confirm it really makes workflow smoother.
- Mike: Infrastructure changes, seeing how that affects all the tests. Now have to ask jgraham or foolip to run through Chromium or Gecko infra. Robert: Philip and I were actually just discussing this. Should have this capability in wpt. Mike: Also have a problem with a copy of testharness.js in Chromium.
- Philip: Seeing the results of Chrome, Edge, Firefox and Safari on PRs, for new tests or modified tests. Would require GitHub-integrated CI for Edge and Safari.
- Luke: Ability to verify changes in a confident way is the high-level of the above requests. Regressions end up downstream due to 2-way sync, taking longer to revert.
- Jon: GitHub notifications. Also console output for `wpt run` is too verbose.

- Sam: Similar to above, and GitHub notifications. Often want to “mark as unread” or something.
- Boaz: As test writer, a sense of coverage and whether a test already exists, or where to put a test. Also how to audit automatic exports from 2-way sync.

Presenting “[An update on wpt.fyi](#)”

2018-06-12

Jon presenting today.

Roundtable Q:

- Philip: 2. new viable browser vendor (people who cared a lot would start it)
- Simon: 5. Visualizing regressions between PR and master. Visualizing results of stability checker.
- Boaz: 1. Web framework bug input. (Bugs in React, that kind of thing.)
- Sam: Pass
- Robert: 2. End of the world. Browsers so complicated that it’ll be a disaster for the open source community to take over completely.
- Luke: 4. Test outcome history grid. What a single test has been doing over time.
- Mark: 4. By-folder release version graph. Want to know that in M65 I couldn’t use it, but now I can in M66.
- Mike: 4. API-based organization. To web devs, only top-level metrics make sense, but at a more granular level they can’t tell what aspects they can use.
- Jon: 2. I’m a big fan of music and music production, love soundcloud. Every month there’s a “soundcloud is going to die” article. Such a niche that if it vanished, something would take its place almost immediately. If the community is strong enough, the hole is filled again. Think that would happen in this case.

Presentation on [Visualizations](#).

2018-06-05

Present: Tab, Boaz, Philip, Jon, Mike, Simon, Sam, Luke, Robert

Tab presenting Bikeshed test annotations.

Boaz: How do you which tests to link?

Tab: “WPT Path Prefix” in bikeshed metadata, plus `<wpt-rest>` or `<wpt>foo/bar.html</wpt>`

Boaz: Reminds me of [what Rick Waldron did for WebRTC](#).

Tab: Documentation of `wpt.fyi/api/` a bit sparse.

Luke: Let’s talk. Currently `wpt.fyi` frontend is only client of API.

Boaz: <https://rwaldron.github.io/webRTC-pc/>. PLH has been talking about redesigning /TR/.

Tab: Would like to have an alternate view for wherever the specs are. New /TR/ URL design will make it easier. Difference from rwaldron's work is blocks vs. linking directly from some piece of spec.

Simon: Warning if <wpt-rest> is non-empty?

Tab: Not now. If you want that, don't use <wpt-rest>, then it'll blow up. <wpt-rest> is a debugging tool.

Simon: Wonder where to put historical.html. Negative tests.

Boaz: Might encourage people to add historical sections to tests.

Philip: <https://dom.spec.whatwg.org/#historical> exists.

Boaz: Luke, planning to update API for wpt.fyi to cater to this?

Tab: Current endpoint gets me what I want.

Boaz: I'm querying the API currently.

Luke: We'd like to be able to query things like "all timeouts" or "all Chrome failures".

Boaz: Scoped results would be great.

Luke: I'll keep you in the loop!

Simon: Luke, would it make sense to use BigQuery?

Luke: We've considered it, will let performance dictate.

Tab: I'll end up wanting all wpt results.

Luke: Sounds like you're a client similar to wpt.fyi itself :)

Philip: Possible to override prefix to link to other directories than prefix?

Tab: Yes, with <wpt prefix=/something/else> or just <wpt prefix> to reset it.

Boaz: Metadata requirements for spec authors?

Tab: Seems like large upfront cost, but after that easy to update.

Philip: Do you think renaming will be a problem?

Tab: If we have that metadata via an API that'd be something I could use. But don't think it'll be a big problem.

Philip: Documentation?

Tab: Yes, will write it today!

2018-05-29

Present: Philip, Simon, Boaz, Sam (in chrome canary), Jonathan, Mike,

Round the room topic: How did you get in touch with WPT? When? Why? Or: If there's something with the project that annoys you.

Simon: Probably aware when it was created, James Graham wrote the bulk of the runner. Before that we had ad-hoc frameworks at Opera. Wanted to share tests. Thing to fix: when someone opens a PR, show results before and after and across browsers. Stability checker, but better.

Sam: James told me to review the test harness he wrote.

Boaz: In 2012/13 when working with core mobile group at W3C, rng.io. Had to use testharness.js. Would want the test harness to be more familiar. And test coverage. And Tobie Langel calculating how much effort it'd be to test the web platform.

Robert: Last year after graduating and starting at Google. One day before starting got email from Philip via RByers about it. Agree with Simon: regressions in PRs.

Luke: Found out when joining, agree on regression detection.

Jon: Like Robert and Luke, joined the team. Change: default output to be way less verbose and look more like when running other test frameworks. I see a wall of text and don't know if tests ran or passed.

Mike: Jon, you can do that with a new reporter. Add a new one, which then becomes the default. For me, I'd like to put tests in a subdirectory. Already been shot down twice :)

Philip: Think I learned about WPT from Simon when working on media elements. What I'd like to fix: regressions and PR processing. Finding regressions on main line runs.

Simon presenting: [Disabled/flaky/slow web-platform-tests Report](#)

2018-05-22

Present: Boaz, Philip, Rick, Robert, Luke, Jon, Mark, Simon, Luke Z

Today's prompt: something exciting in the last week or so.

Boaz: Mike's proposal for virtual device drivers in WPT. And the ARIA WG testing effort.

Rick: Confluence graph at Google I/O. People are asking us how interop is doing! People are asking specifically about web-platform-tests.

Robert: Have been working on results receiver, got to work golang for the first time. Ready soon.

Luke: Excitement on wpt.fyi progress on 3rd party runs. Also being a CI, detecting regressions, that kind of thing.

Jon: testdriver interest, working with various people on testdriver additions.

Mark: Exciting to see some Go infrastructure working as intended. Metrics computation seems to be running smoothly now.

Luke Z: Also in WAT, learning about the team, maybe joining in Q4. Now work in Chrome security.

Simon: Adding OWNERS files to everything in web-platform-tests. Lint check as well, Sam wants tests for the linter :) Also disabled tests report. Fixed a bunch of tests. Split some big tests so they don't take too long.

Philip: the tutorial for adding stuff to test driver was exciting, and I know people are extending test driver as they work on implementation instead of waiting.

Philip presenting [The Interop Machine](#) today.

2018-05-15

Present: Boaz, Mike, Sam, Simon, Robert, Mark, Luke, Jon

Skipping the round-the-room because Philip didn't think of a topic.

James presenting on [Taskcluster](#) today.

Discussion:

- Sam: Limited macOS capacity. What about Windows?
- James: It's cloud-based, and if we wanted to use it I could find out how to make it work. Might make sense for CI (PRs). But Edge Insider Preview might be difficult.
- Philip: Size limits for artifacts? Could we store 27k screenshots for retests?
- James: Don't know. If you use too much storage someone will probably complain.
- Sam: Also don't need to store redundant screenshots.
- Mike: Distinction between GitHub integration and using it programatically. Any reason to use it programatically?
- James: Decision tasks would be programmatic, and then the .taskcluster.yaml would just schedule the decision task. In Gecko, the decision task itself reads a bunch of YAML files. But that's not directly reusable.
- Robert: With a task graph we could also do results merging.
- Philip: And detecting regressions and failing PRs?
- Robert: Yes.
- James: Can schedule things that depend on more than one browser.
- Boaz: Docs say that Taskcluster isn't easily redeployable. Roadmap?
- James: Memory is that it was deprioritized initially, but now more possibility of that happening. See <http://code.vigoro.us/posts/2018/01/r14y.html>
- Philip: Debugging? Can the "timeout at end of run" problem be debugged without special powers?
- James: Hasn't been debugged, but yeah. [showing where to find things in UI]

2018-05-08

Present: Mark, Luke, Robert, Jon, Mike, Simon, Dru, Philip, Boaz

Dru presenting today, in final meeting as team PM, today is last day. Transferring to search in Tokyo.

[“The Predictability Carol”](#). Past, present, future structure to talk, like The Christmas Carol.

Discussion:

- Jon: Writing a new web engine. Very hard?
- Dru: Yes, the difficulty is only growing. Another dynamic is that when market share drops, interop problems become compat problems. But WPT gives you something that allows you to be proactive, lowering the tipping point of market share required. We’ve guessed 8-10% before, WPT should lower it. But from-scratch impl still hard.
- Mike: Seems less and less feasible to start up a competitor. A bit like Linux. New browsers are all Chromium, just like all distros are based on Ubuntu. Serving people who only want part of the web would also serve people who are building new browsers. If truly composable.
- Dru: The more modular browsers are, the easier it is to reuse just parts of them.
- Philip: Forkability really important. WebKit and Blink started as forks.
- Mark: Composability on the web platform as well. Encouraging web developers to match the composition of browsers, so that parts can actually be removed.

Philip: Thank you Dru, a please to work with you. Recommend the [Interop Roadmap](#) for everyone who wants more of Dru’s wisdom.

2018-05-02

Sam presenting today.

Question: what browser bug surprised you the most when you found it?

- Boaz: Don’t know... the one with clipping disagreement between Firefox, Chrome and Safari. Surprised at disagreement between following the spec vs. unbreaking 100s of sites.
- Mike: overflow-x and overflow-y, how they interact when you set them together. And that it’s still unresolved. Spec isn’t really clear, nobody seems committed to following through.
- Robert: Font matching on Chrome for Android. Seems like it falls back to sans serif even when you ask for monospace in surprising ways.
- Jon: How DevTools shows promises when logging. Order is different in different browsers and it’s very confusing.
- Luke: When working on blogger, had a problem with a Google font on Edge, that dir=rtl made a whole subtree fail to render. Took as a day to work out the conditions were that broke that. Combination of nested divs, a particular font and text direction. Another was when doing parallax effect, propagation of scroll events doesn’t work with non-body root scroller. Whole page would lock up after scrolling for a bit. Probably fixed.

- Mark: a JavaScript 1-liner that could crash Safari. Accessing members of the global objects using the prototype and the wrong object would crash everything. Was fixed after 6 months.
- Philip: I don't have good answer. Sorry.
- Simon: Before the HTML5 parser was implemented, hitting EOF while in a comment, all browsers would rewind the input stream and reparse from the beginning of the comment, making something that should be in a comment suddenly executable code. Now fixed!
- Sam: Opera's JS engine Carakan. If you shifted -1 by 0 bits you got INT_MAX or something. Broke a popular blogger theme and only found after shipping :)

Sam presenting on wptrunner. Slides not working, but main slide is:

http://wptrunner.readthedocs.io/en/latest/_images/architecture.svg

- Some additional background information about sub-processes created by the WPT CLI: <https://github.com/w3c/web-platform-tests/issues/9551>

2018-04-24

Just doing a standup this week, no presenter due to a late change of plans.

- Sam: adding extra hosts to WPT, a bit of a debate about how to refer to them syntactically.
- Mark: Splitting time between improving Confluence (going well, building on MDN work) and wpt.fyi endpoints for revision announcer. Deployed at <http://revisions8-dot-wptdashboard.appspot.com/api/revisions/latest>. Also turning that into PRs.
- Mike: [`nonexistent-origin` exists in GNU/Linux](#). But the tests that rely on it just use it as a way to get a bad request, and they get it because of cross-origin issues, so working for the wrong reason.
- Simon: Triaging test quality issues in WPT that would be valuable to fix. Highest prio is a list of tests that are disabled or marked flaky in more than one browser's repo. Have also sent PRs to add OWNERS files, starting with adding spec editors in css/. Also looking for implementers.
- Philip: spending too much time on <https://github.com/foolip/ecosystem-infra-stats/issues/3> and <https://github.com/foolip/ecosystem-infra-stats/issues/11>, unable to rely on GitHub API.
- Robert: 2 things. (1) is the results receiver where design doc has been sent out, just need a few details ironed out. Will deploy to staging. (2) is to support FindIt, which is a Chromium tool to identify flaky tests, and to teach it to find flaky LayoutTests.
- Jon: Adding more things to testdriver.js and documenting all of them.

Robert: Simon, on your lists of disabled/flaky tests, do you have a tool to get a list?

Simon: not yet

Robert: for TestExpectations, you can probably use a regex, but if you need help reach out to me.

Simon: Wrote a grep for more than 1 token in [here], and also slow tests are interesting.

2018-04-10

Mark running the meeting today.

Attending: Mark, Luke, Robert, Jon, Philip, Boaz, Mike

Around the room: What is most broken about the ecosystem built on the web platform, what incentives led us down the path of the thing that is broken, and what should the incentives be so that we don't do the broken thing?

- Robert: Don't understand why JavaScript has so many libraries and frameworks. Sheer number of npm packages amazes me. Python is a similar kind of language, but JS has much more packages. And toolchains are complicated, multiple build and testing tools for a single project. I'm not sure why this happened. I think it's because there is no standard library, and huge use. Python has PEP, where in people can propose changes. But JavaScript is much harder to change.
- Boaz: Don't know if it's incentives problem, but one thing we've talked about is to standardize the node global, which together with the web platform could make up a standard library. And think that the ecosystem is very male dominated, but don't think we can just create incentives to fix that, but do other things.
- Luke: Hard to isolate one thing. The theme is that there isn't a formal monolithic way of doing things. IDL isn't centralized and dependencies are a problem. Weird that the web doesn't have a similar package management system to node. A more standard way of submitting packages and declaring dependencies might fix it?
- Jon: Yes, too many guys. Think that the node packaging problem is both a good thing and a bad thing. [foolip lost track scribing] More consistency among frameworks and the JS being used would be good to stop the framework skills churn.
- Philip: I wish I could say that what our team is what's most broken. The ads ecosystem is pretty broken. It wants user attention, but it doesn't seem like it's pretty broken, and I don't know how to fix it.
- Mike: Maybe identity, and that we still don't have a solution for that. Forces us to make compromises in how we interact with stuff. Try to push back against Facebook and GitHub auth, but feels like it's impossible because we don't have federated identity. Bruce Schneier talks about feudalism, and how we're in a similar relationship with service providers. Don't know how to fix it, can just complain.
- Mark: Very hard to fight network effects. My contribution: a lot of smoke and no fire around progressive enhancement. Bothers me because we end up with a JavaScript-heavy web where it's supposed to be the add-on to the web. Wish there was something like react or angular that produces something that looks good, but different across clients. Doesn't have to look the same, would lead to a more inclusive web.

Mark presenting the [revision announcer design](#).

Discussions:

Philip: Mike, what is the strategy the infrastructure is currently using?

Mike: Right now, every 6 hours we fetch from Git and runs the latest commit, for Chrome and Firefox. Every 48 hours we do the same for Edge and Safari. So we have a microcosm of the very problem that Mark is trying to address.

Philip: Guaranteed that the 6h and 48h will align?

Mike: No, not technically guaranteed, there's a 5s delay or so.

Mike: Had feedback from James Graham on the list about using Git as the database. So that runners would just be fetching tags and maybe special branches.

Mark: A drawback with that approach is that to change the collection of frequencies, you need to (obviously) change the announcer, and also find all the runners and tell them about the new cadences.

Mike: Aren't we trusting the runners to pick the appropriate cadence for them?

Mark: Yeah, but if runners run continuously, then if they just run as fast as they can, then by this scheme improving they'll get onto the faster cadence just by decreasing the running cycle time.

Mike: Can't runners have logic like this, with special branches?

Mark: Yes, runners can do something isomorphic, but thought that's extra complexity as well.

Mark: out of time, but sounds like Mike and I should sync about tags in git.

2018-04-03

Luke running the meeting today.

Attending: Mark, Luke, Jon, Robert, Boaz, Sam, Mike, Philip

Around the room: what you're expecting to work on in Q2. when dealing with our own OKRs we might not know what everyone else is planning.

- Jon: fix the Travis CI stability checker timeouts, using some heuristic to run fewer times. Expand the automation capabilities of wpt. WPT documentation. Implement permissions API in ChromeDriver.
- Robert: Microservices surrounding wpt.fyi, including endpoints to receive results from the runners, work with Mark on new storage format for results, and keep import/export running.
- Philip: rotation running smoothly, making that web app suck less over time, transition WPT to a new github organization (which includes talking to other engines to not break things), work on upstream blink infra a bit (wpt infra roadmap issues).
- Sam: wpt roadmap issues, not sure which subset. Get more tests that are being written into WPT, both in Blink and Gecko.

- Boaz: Separate wptdashboard repo, search/investigation of completeness on target platforms, collect bugs and fix them in wpt run and WebDriver implementations. Life support for pullresults. If there's time, "turn up the volume" to run more frequently. Also looking at interop spec bugs.
- Mike: What Boaz said. Question: are there stable URLs for Chrome builds for Linux? Want to be able to include the browser version in the results to be able to reproduce.
 - Mark: We probably don't do that by design, so that there's no public URL for old versions (avoid version locking and promote good security practices)
- Mark: Continue to fix things in Confluence for a broader release, work with Robert to design/implement coordination of runs, the list of important hashes we provide to all runners. Also make the tool for updating MDN's browser-compat-data using Confluence data. Also stretch goal for Chromium's IDL not diverging from the specs.
- Luke: working on most of the things that have been mentioned. Working with Robert on submitting results, and that relates to the work Jon is doing on Travis. And following the long tail of IDL changes in WPT. Some promising automation that Mark has pointed out. For most of the tests I've changed, have found idlharness.js bugs.

Presentation: [WebIDL: Combining spec fragments with sticky tape](#)

2018-03-27

Robert running the meeting today.

Attending: Mark, Jon, Robert, Philip, Boaz, Sam

Around the room: if we were to rewrite wptrunner from scratch, which language would you chose, or definitely not chose, and why?

- Mark: needs to run tests in the browser... is it responsible for managing more than one browser instance?
 - Robert: background. The runner generates a list of tests and sharding/chunking and a lot a process management, mostly talking to WebDrivers or Selenium, which will spawn browsers.
 - Mark: I'd chose a language that's good at concurrent programming, or a continuation-style programming. Go can be good, some magic in C# too, but would stay away from JavaScript and Python as they don't do concurrency well.
- Jon: Haskell! JK :) Go would be a decent choice, but skeptical because we're Google. Mark: Rust? TypeScript? Jon: Maybe Rust or Go, they seem to lend themselves well to the problem. Python 3 is the other obvious choice, which we shouldn't dismiss. So Rust.
- Robert: I'd want to stay away from Python 2, because it's going away.
- Boaz: JavaScript, Python or Perl. Perl because it's used in WebKit. Rust also came to mind.
- Philip: JavaScript (node, with typescript), it doesn't matter that its single threaded because of childProcess. It has to be easy to get running on all these platforms. If we

can't get it running on Linux, Mac and Windows it's no good. So what works everywhere? Mark: Java? Python3, JS/Node/Typescript.

- Mark: lots of people will be running this code, but how many people do we anticipate contributing to wptrunner itself? Jon: the number of people who contribute depends a bit on how ergonomic it is.
- Robert: want to clarify that this is still hypothetical :)
- Sam: Rust. Because of concurrency, and Rust has a type system that can enforce type safety.

Robert presenting "[WPT-RESULT-HUB \(WIP\)](#)"

Questions:

- Boaz:
 - Push or pull to or from "results hub"? I know that Mike from Bocoup has a pull based proposal.
 - How important is it that this is reproducible outside of Google?
 - Robert: would like to start with AppEngine TaskQueue, but maybe have an abstraction around it.
- Sam: Why extend wptreport instead of using one of the other output formats?
- Mark: could we try to put all of our extensions in headers (out of band)?
- Robert: Reason it's based on wptreport is that running infrastructure now generates that. But we could chose another format or modify a format.
- Sam: If 60s is a problem, maybe just use some faster compression format?
- Mark: If we can get the data into some storage in 60s, we can handle the request in that time, but do extra work afterwards.
- Sam/Robert: also prefer this.

2018-03-20

Jon running the meeting today.

Attendance:

Philip

Robert

Jon

Luke

Sam

Boaz

Prompt: Suppose we had all browsers and all runs reliably, what would be next for the dashboard, what features would you like to add and should they take priority over the run priority that we've established. Answers:

- Jon: a subscription service to notify browser engineers of regressions, so that they can see them on wpt.fyi. An email would be sent to me.

- Robert: I was going to say the same thing :). Instead, I would add: a mapping between specs and tests would be useful, to filter tests by specs for example.
 - Luke: would we have if we had a consistent directory structure?
 - Robert: that directory structure seems like a stretch.
 - Luke: I think the mapping should be more rigid than what we have. Tab's working on a feature, and being able to determine "code coverage" is a feature I'd be keen on. Requires mapping both ways.
 - Jon: Rick said last week that code coverage has been OK in Chromium because there's been a culture of testing, but the same isn't true for the web platform.
 - Sam: I'd like to get from a failing test to a bug. Philip:
 - <https://github.com/w3c/wptdashboard/issues/30>
 - Jon: wouldn't be hard to iterate all failing tests and robo-filing bugs. Everyone: noisy. Jon: Yes, this is a horrible idea, it's just the simplest thing that could work.
 - Luke: Sam, when do you think this could become useful? Does it require the number of failures to be very small?
 - Sam: with many failures it's infeasible, but you could imagine marking tests that have been fixed in stable. But that might not be so useful anyway.
 - Boaz: (1) per-browser reports (2) mapping between specs and features, making it useful for web developers
 - Philip: We should not just be testing master, but we should be testing PRs and doing useful things with those results to figure out which PRs introduced failures, and [Surface test regressions from Travis as review comments](#).
 - Luke: said mine, but having more rigid 2-way mappings between tests and specs.
- Jon presenting "[Hi, my name is ecosystem infra](#)"

2018-03-13

Skipping the rounds since most of us have been talking for hours.

Sam has been breaking stuff and then fixed it again.

Mike Pennisi presenting today: WPT, Amazon & Sauce Labs

- Slides: <https://bocoup.github.io/presentation-wptd-parallelization/>
- Source: <https://github.com/bocoup/presentation-wptd-parallelization>

2018-03-06

Philip presenting

Prompt: Biggest frustration with WPT or a web spec.

- Boaz: realms and how to deal with realms between JavaScript and the web platform. Philip suggests that domenico@ has the answers.
- Luke: "Excited" about the lack of rigidity around line height, have struggled with that a lot.

- Sam: The fact that so much of CSS is undefined, that table layout is undefined. I mean it is!
- Jon: A lot of behavior around fonts, when to fall back to sys default, when not to. Especially on android.
- Mark: Scrolling using flexbox scrolling requires a non negative min-height. Why?!
 - Sam: this comes from an old version of the spec
- Robert: The whole font selection/matching thing has just too much complexity between browsers and systems. The whole thing is especially bad with CJK fonts
- Philip: WPT run is fairly complicated and trying to understand why it hangs is difficult. When it hangs and you kill it, it doesn't kill all subprocesses

Philip presenting on [wpt.fyi running infra](https://wpt.fyi/running-infra)

2018-02-27

Tab: What is your favourite language feature that you wish everyone would use?

- Jon: Map and filter
- Luke: integrated built in querying, sorting (e.g. Linq in C#)
- Tab: observables, (dom is trying to fully bring to js).
- Robert: dynamic languages that have great support for typing
- Mark: go routines and channels

Present:

Tab:

- [Bikeshed inline testing feature](#)
- Working on this off an on
 - V1 is up
- Maintaining test suites for specs is hard work
 - coverage
- Take an element called wpt and write the test path for wpt next to the part of the spec you are testing
- Easy to check coverage, easy to see what needs to be updated when things change
- Tests don't show up by default in the compiled bikeshed version, but you can make a test annotated version of it.
- Wpt-rest prefix
- Future work
 - Want to add live test results
 - Test Index on all specs
 - Wpt.fyi api use
 - <https://github.com/w3c/web-platform-tests/releases/latest>
 - Warn when test exist that aren't listed. Problem: some specs share a folder
 - Test name templates

Discussion

- Jon: What about linking existing wpt tests that have no wpt elements inline in the spec and notifying people that this is the case?
- Robert: The manifest is available as a gzipped file from the releases page

2018-02-20

Present:

Q1 OKR Doc:

https://docs.google.com/document/d/1ojLLDiRul4e95Z7JQ10JQPRMiQHd_40rBkYCL6IJgw/edit#heading=h.qby8oq1ilp8

OKR Midq status:

- Bocoup (Rick, Matt, Mike, Boaz, Marie)
 - O: web-platform-tests dashboard (wpt.fyi) is reliable and useful
 - KR: Changes landed in wpt are tested and shown on wpt.fyi within 12 hours (P1)
 - KR: wpt.fyi always shows results for the same wpt commit (P2)
 - KR: Make experimental browsers the default configuration (P1)
 - KR: Changes in wpt are tested in stable browsers within 7 days (P3)
 - O: web-platform-tests continuous integration is reliable and useful
 - KR: wpt PRs have results from Chrome, Edge, Firefox and Safari within 15 minutes (P1)
 - KR: Surface actionable test results as GitHub review comments (P2)
 - KR: wpt CI runs the same experimental browsers as the dashboard (P2)
 - KR: Complete a detailed infrastructure design for deduplicating / sharing data between PR results collector and general wpt.fyi runner (P3)
 - O: Upstream web-platform-tests infrastructure is well maintained
 - KR: Overhaul web-platform-tests documentation (P3)

2018-02-13

Present: Philip, Danyao, Luke, Mark, Ali, Boaz, Joe, Sam

Presenter: Ali, Danyao

Prompt: Apropos Ali and Danyao's attendance, what one thing would you like to see in Safari or as part of the WebKit project?

- Danyao: One thing I would like to see would be goma incremental compile tools
- Luke: pass
- Mark: I found a bug where if you put things on the window object it would crash, so I'd like to see better isolation so that the whole browser doesn't crash.
- Ali: Better documentation. Sometimes look at Chromium's docs for things in WebKit :)
- Boaz: better compile workflows and documentation for JSC and hosted per-platform builds. And 2-way sync for wpt.

- Joe: Service worker parity.
- Philip: PWAs and add to homescreen, especially if they could one up us and put us in their app store the way MS is doing. Support for the same codecs as everyone else, which is a problem for webRTC where everyone supports vp9, and webkit only supports h.264.

Ali presenting on [web-platform-tests on Safari and Chrome on iOS](#).

Discussion:

- Mark: how much work would it be to not restart the browser as much?
- Sam: there's some logic to restart on unexpected results, and also after every 100 tests or so.
- Where is it configurable to not run in a new browser each time?
 - `./wpt run ... --no-restart-on-unexpected`
- Luke: You used a patched version of wptreport. wpt.fyi also has an endpoint to diff runs. Have you considered posting these results to wpt.fyi? Ali: sounds great!
- Sam: the WebKit debugger has some way of attaching remotely, and some wrapper around it. Maybe you could automate retests using that?
- Danyao: in the simulator there's a way to attach, interesting idea.
- Ali: there's a thing called appium, but unable to dismiss the popup warning
- Mark: would be great to have strong tooling for running in parallel, for example being able to run using BrowserStack with multiple instances in parallel.
- Boaz: My coworkers are working on running wpt in parallel.
- The server hosting w3c-test.org is a small machine, and they don't intend for it be used that much.
 - Ali: we were running it on a local server, and were getting 400s. Boaz: Can you file an issue on that repo? Ali: Yes!
 - Follow up: Ali was not able to repro.
- Philip: we have [Chrome for Android](#) support now for the purpose of comparing with UC and QQ (based on Chromium) which is a similar kind of project.
- Danyao: we're not able to compare to safari TP
- Philip: Bocoup is working on that as a P2 for the quarter

2018-02-06

Present: Mark, Luke, Philip, Boaz, Sam, Joe

Leader: Sam

Prompt: if you could change one thing about the web platform with no regard for backwards compatibility, what would it be?

- Joe: https only, across the board
- Jon: make it such that it was the standard instead of apps, so that apps were never a thing

- Luke: it would be good if we could leverage p2p as a way of reducing the amount of downloading that needs to happen in a way that is common across websites
- Mark: layout object model for viewports, not documents. The VOM instead of the DOM.
- Boaz: get rid of window in favor of a standard library for JavaScript that could be shared across other JS contexts as well.
- Philip: I had some time to cheat by looking at chrome status dot come
 - <https://www.chromestatus.com/metrics/feature/popularity>
 - Sync xhr is horrible
 - Cookies are so terrible, they should follow same origin policy
 - Nuke the public suffix list
- Sam: similar to Mark's thing. So much is applications now, and CSS makes less and less sense. Replaced with something more like what Android and iOS do -- a notion of what's on the screen, things are only scrollable explicitly, etc. etc. Mark: there are fewer layout modes on mobile, and if you need a different one you write code for it. Much simpler to reason about.
- Mark recommends [The Fucking Open Web](#) and [The Open Web. Fuck Yeah!](#)

Presentation: [Automation: the now and the future](#)

Robert's WIP for UC: <https://github.com/Hexcles/web-platform-tests/pull/1>

2018-01-30

Present: Mark, Boaz, Philip, Robert. Mark running the meeting.

Prompt: Aspirational roundrobin of share an existing or not yet existing WP feature to improve the lives of **browser users**.

- Robert: Not a front-end developer even though I did an internship writing CSS and HTML. But based on that, I think **web components** are going to be very useful. If developers have an easier time writing that stuff, more time for what matters.
- Boaz: Re web components, the "view source" culture comes back. But otherwise, WebRTC and P2P features that enables a kind of privacy that's good for users.
- Mark: What I think would make users lives better is if there was a pervasive cross orgiin mechanism so that we can build decoupled systems of information and interaction so you can build lots of types of systems and use that as atool on lots of different things.
- Philip: I am in agony, this should not be a hard question. Text on the web is very inconsistent, but I don't know how to fix it. Browsers are building "reader modes" in. This is in conflict with ads, so site developers don't use them.
 - Boaz and Mark: AMP?

Mark presenting some of the things that Joe Medley and he did last week, using confluence data to vet MDN data: MDN Compat Curator Support Tools

- [PRD link](#)
- [Presentation link](#)
- [Paris Hackathon March 16-18](#)

Questions:

- Boaz: This is great, kudos on resisting the urge to automate everything, smart! And what's the search language?
- What was the search querying? Mark: I used FOAM
- Robert: the search feature would also be useful for confluence. Add it there too?
- Mark: Yeah, I hope that the confluence version of this can replace the current API catalog. Part of a Q1 OKR about surfacing search parameters.
- Philip: You started this for confluence. Mark: sort of, hoping to switch it over.
- Philip: I agree that resisting automation is very smart. We had the same in two way sync.

2018-01-23

Present: Joe Medley (visiting WAT), Robert, Luke, Raphael, Boaz, Mark

Boaz: Marie and Matt won't be joining these meetings, switched things up a bit.

Robert leading the meeting.

Topic of the week is a recent challenge.

- Robert: Q1 plan has been changing a lot. Unexpected infra issues come up a lot, and a surprise visit to China, which I'm excited about and preparing for.
- Mark: the complexity of mashing up an elaborate scheme -- working with Joe to work out what would be most useful for updating MDN's compat data. A lot of structure to that data, not trivial to link it up with Confluence data.
- Luke: extracting pieces of web components. Also had a bit of fun (challenging fun) with the remote API in app engine. And Docker!
- Joe: discovered that the Feature Policy documentation is going to be a lot more complicated than I thought.
- Philip: compat experiment for gecko had me pushing on [data url parsing issue](#) and have been having a fight about whitespace, and in these negotiations we can't come to an agreement with other implementers (apple) the work is dead in the water, so my challenge was doing the research and presenting the case without being a jerk. We ended up in a good place and annevk updated the spec with what I wanted.
- Boaz: Figuring out how OKRs intersect with other roadmaps. And creating a product roadmap for wptdashboard. Kickoff meeting went well. Also working with other implementers, and helping them "catch up" on things like 2-way sync.

- Raphael: I've been away for a bit. Biggest challenge is getting back to the infra work and balancing that with the other Intel work.

Mark: Boaz, please reach out if we can make the data from wptd, confluence and IDL infrastructure more useful to other vendors.

Boaz: and this week is TC39, watch out for changes to JS!

Robert presents: [Chromium WPT Sync](#)

Questions:

Boaz: In export, how is it blocked by Travis? A benefit is that people can notice problems in upstream wpt before being merged in Chromium, but how does this work?

Robert: Most of the time, Chromium devs don't rely on the fact that a PR is created quickly. But if a CL lands without passing wpt's CI, and then we have to fix the change in wpt manually. Sometimes Mozilla has to merge a test as flaky in their manifest.

Boaz: The 30 minutes that it takes Travis to run, can this be a blocker for landing a CL in Chromium?

Robert: No, that doesn't block Chromium's CQ. The exporter just tries to merge PRs every 5 minutes.

Boaz: we are going to be working over the next few quarters on tooling for import in WebKit, mostly Test262 but also a bit of WPT.

Boaz: Can I share these slides, can you talk to people working on WebKit and EdgeHTML?

Robert: Yes!

Mark: Can we please have Edge's IDL files? :) The auto-updating of IDL in wpt will be more valuable down the line if we have everyone's IDL files.

2018-01-16

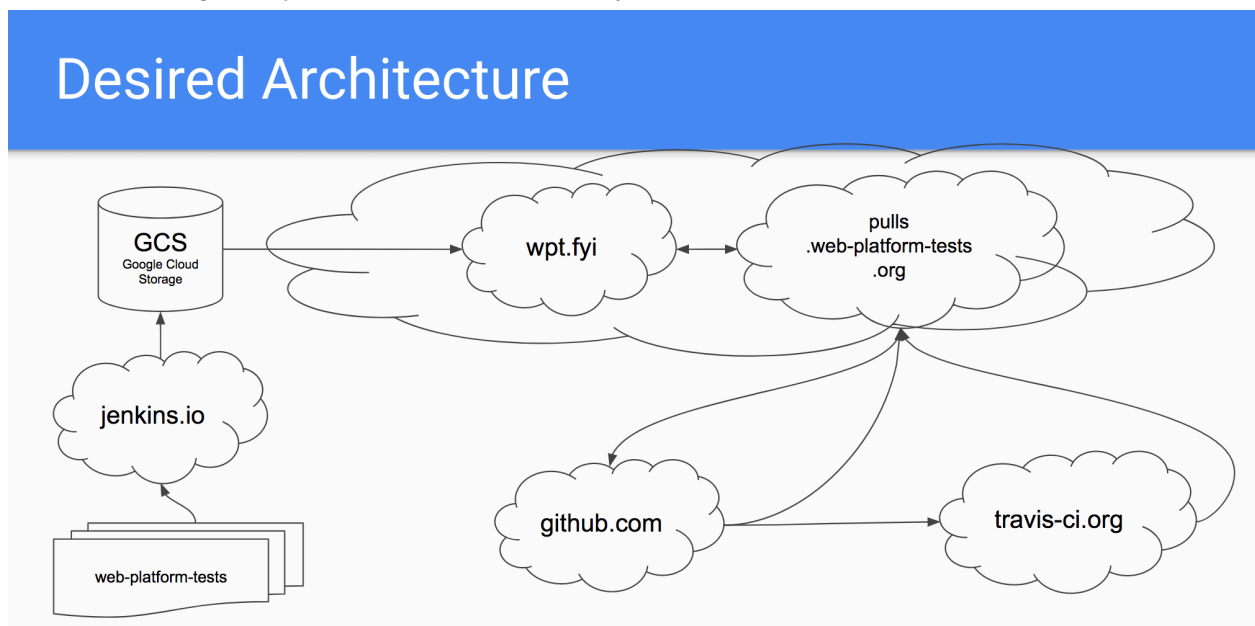
Luke running the meeting today. At the end of the meeting, will present on the dashboard work from last quarter, that also might change the architecture a bit.

This week's question: what have you been doing / hope to be doing?

- Joe: IDL databases up and running on my machine. This week making sure it works remotely, going to WAT next week. Also wrote a bit of PRD.
- Boaz: Worked on deployment issues for wpt.fyi. So now I know how that works. Also internal planning on the Bocoup side for how to divide the tasks, which continues.

- Philip: Last week ran confluence against UC and QQ. Robert and I are going to meet with UC and QQ in their offices in china. I'm also supposed to be running those browsers against WPT.
- Sam: Reviewing PRs in wpt, some people were productive during the holidays :) Now working on not-w3c-test.org stuff to test things around cookies.
- Marie: Doing some of the same things that Boaz mentioned. Have talked in a lot of detail about Q1 OKRs which has been very useful. That planning this week, and onboarding Mike Pennisi in February.
- Matt: Stabilizing the running of wpt.fyi. And lots of meetings ahead of kickoff meeting. Happy to help review PRs, please ping :)
- Luke: Pulled apart the metrics prototype into smaller bits that are easier to review. And wrote an anomalies view. Also need to reduce the data size. The interop view is a list of all the tests, with 4/3/2/1/0 columns. The anomalies view are the tests that are passing in some browser but failing in some.
- Mark: I worked on confluence and wptd. Worked alongside Luke to get the metrics code reviewed and landed. In confluence, I did mostly supporting others (Philip and Joe) and then adding docs from what I learned.
- Robert: I spent some time on blink internal infra, trying to get layout tests in blink working on newer systems, especially Mac High Sierra. I worked with Jon to get chrome on android with WPT, which surprisingly, no one has done yet.

Luke presenting [wpt.fyi: A feature overview](#). Key slide:



2018-01-09

Philip will talk for 5 minutes

- We're experimenting a bit with who runs the meetings so that I (philip) can participate.

- We got a .36 for our quarter, which is a bad score, but I thought it would be worse. We set aggressive OKRs, fell short, and that is OK.

Jon will talk for the rest.

The round-the-room topic of the week: What would it take for your project to be considered finished, or if cannot be finished, how come?

- Jon: I work on automation on the web and in wpt. The closest to complete would be if anyone who wants to write a test in wpt can easily automate everything without asking, because there are examples and good documentation.
- Robert: 2-way sync. Might not be finished any time soon, mostly because it's a moving target. The actual code is relatively small, but it's glue code talking to different pieces of infra. And the infrastructure itself is changing rapidly, with changes like swarming. So we have to change the code accordingly. But I expect the required maintenance to be lower and lower over time, with fewer and fewer edge cases, and that would be success.
- Luke: dashboard work, surfacing browser-specific metrics, and also updating IDL in wpt automatically. The dashboard work is complete when wpt.fyi is the "first port of call". Should be a useful tool for the browser you care about. IDL project is low key, removing some manual work.
- Mark: Confluence. Done when we have a release process for collecting new data from desktop and mobile browsers that can be handed to any team member, so that anyone can handle "hey, there's a new version of UC, update the data."
- Boaz: At a high level, supporting Bocoup's work, I'm not usually an individual contributor here. That's never done. But making it such that wpt is always up to date, and that every browser vendors are using test results to improve interop / prioritize their backlog.
- Marie: Project manager at Bocoup, so making sure that no balls are dropped. That everything we want to do finds a person to do it. Haven't really thought about an end point. But maybe when every question that pops up easy finds its way to the right person/place.
- Joe: Have set up confluence on my machine now. Found that the README didn't quite work, will submit a PR. Want to visit WAT to jump start the process. Being done means I can type a command and from that I get JSON files that I can merge into the MDN compat data repo.
- Matt: Make sure that everything runs on time and consistently. Should be current and good all the time.
- Philip: In the area of testing in the standards process. I'm done when everyone who works on standards sees testing as part of their job, and that when mistakes are made, when changes are made that never materialize, that those issues are revisited in a timely manner. And that none of this requires poking, but just keeps running for as long as the web is alive.

Jon's slides title [Beyond WPT](#). Would be a good movie title.

Cool idea: subscribe to things being shipped in some browser

2018-01-02

Philip will score overall Q4 OKRs by next meeting. Please score individual KRs by then.

Roundtable topic: most exciting thing we could achieve individually or together in 2018:

- Robert: Get more people using WPT more often. We still see blink engineers using non-wpt layout tests, there is some inertia using old infra, which I think we can fix.
- John: I went salsa dancing with a friend, she lost her phone, and it only worked in safari, and so we couldn't use my android. I want to fix that. Another friend of mine told me they hate writing front end tests with selenium, and that they only care about testing in chrome. We don't want either of these things. So I want browser engineers to start testing in other browsers.
- Philip: Solve the problem of testing and say we are done with that, and move on to doing stuff that is more directly in service of web developers, i.e. have MDN automatically up to date with our testing data. Make cross browser testing easier for web developers.
- Boaz: Over the break, had wpt.fyi breakage on my mind. Ultimate vision is to have wpt.fyi infrastructure be part of all browser implementers work so that they can prioritize the interop backlog. To help improve the workflow of Edge, WebKit and Gecko. Also use wpt.fyi results and perhaps test262 to help web developers. There's MDN, but also other tools that web developers already use. Also excited to run downstream forks of chromium/webkit/gecko in wpt.fyi infra.
- Joe: I'm here for MDN data. I have been looking to automate compat data for use with MDN. And MDN has been moving away from hand built tables to a auto built format based on a json format. This is what I'll be working on in this group. This would make MDN data the only automated compatibility data. I'm supposed to work on this with Mark.
- Marie: I'm generally excited. I love what I've learned about WPT as a project so far. I'm excited to dig into the organizational complexity of how to get this work done.
- Luke: I'm generally excited about delegating in a more meaningful way (to bocoup for infra) and putting my work efforts in more isolated silos. Also excited about working on the MDN automation.

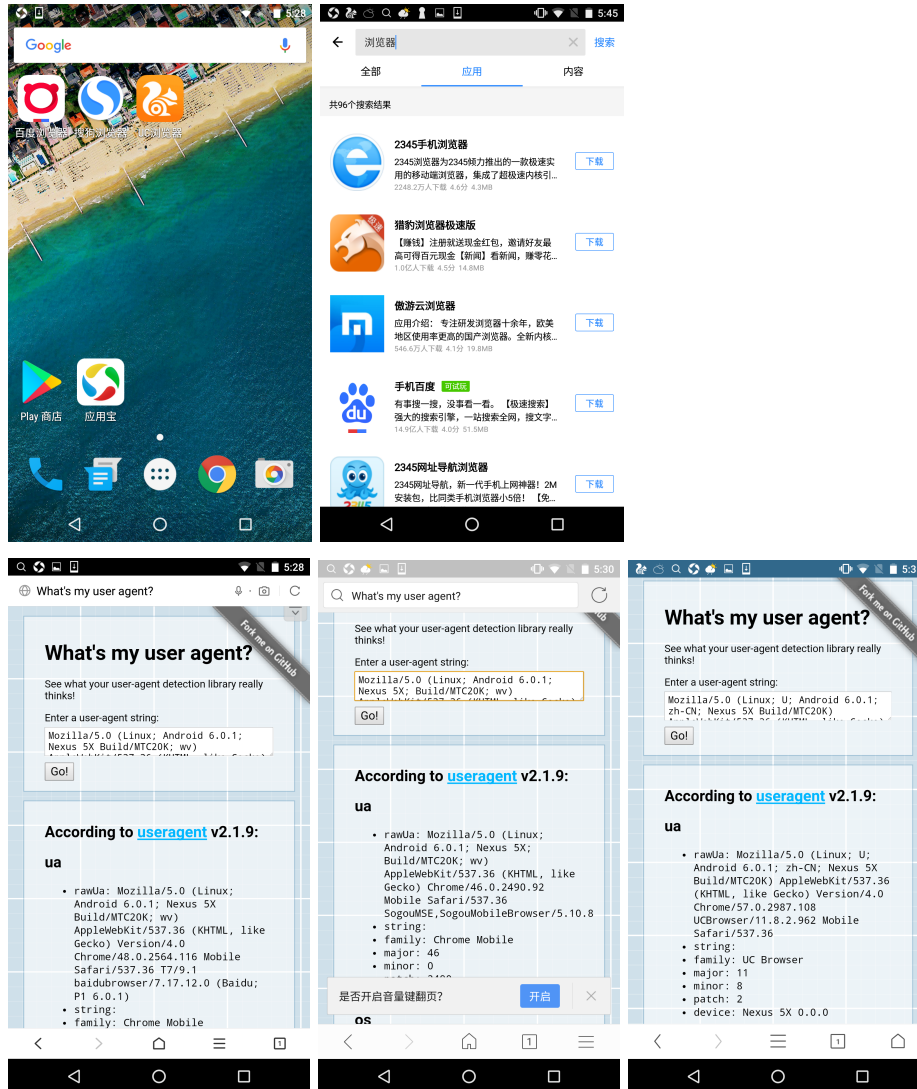
Face to face: some time in February or March in Toronto or Waterloo. (Note: it's VERY cold now.)

Presenter: Philip on getting "other" browsers into Confluence and wpt.fyi.

- UC Browser, Samsung Internet, Opera, QQ Browser, Yandex, etc. All browsers with significant usage in certain regions (or globally). A more regional example is the Vietnamese Coc Coc, Chromium-based.
- Confluence: stable versions. For Chromium-based browsers, ones that can be compared to to Chrome or Chromium built from the same version.

- wpt.fyi: to be useful to the browser developers, need to test the most recent. But in order to compare Chromium-based browsers, stable versions would be more practical.
- How to run tests when no WebDriver exists?

Screenshots:



Additional thoughts sent in [Dump of info/thoughts on including more browsers in Confluence/WPT](#).

2017-12-12

Attendees:

- John K
- Philip J
- Rob H

- Matt L
- Boaz S

Going through [Q1 OKRs](#) person by person.

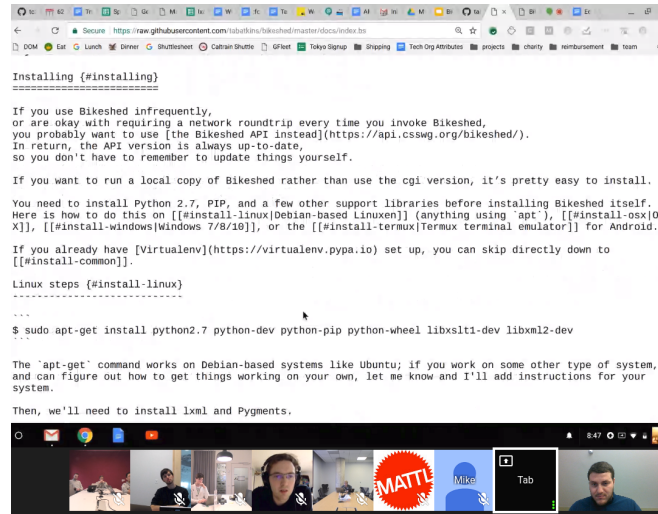
Missing:

- Flexbox work (Later: covered by spec coverage work, but no upstreaming efforts in Q1)

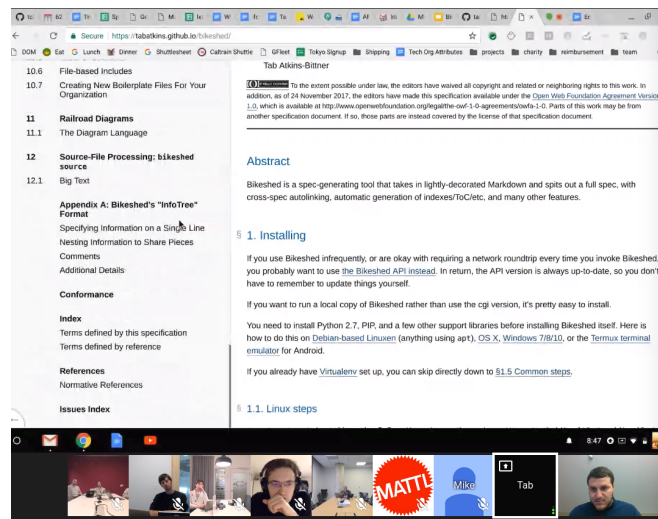
2017-12-05

- Philip
 - Today Tab is going to do a 15 minute presentation about Bikeshed
 - <https://github.com/tabatkins/bikeshed-data> <- Bikeshed's cross-spec linking DB
 - <https://tabatkins.github.io/bikeshed/> <- Bikeshed's docs
 - https://docs.google.com/presentation/d/1tF2dqykVGIM2RpRVhfSnqw4OYbQ3jDZbMqm9J7xSOs/edit#slide=id.g1df26000aa_0_58 <- Some more advanced tips and tricks for using Bikeshed
 - Today I'm suggesting that we skip our round robin to do our OKR planning work
 - We need to have our q1 okrs done by dec 15th. Everyone will go off to suggest some KRs, next meeting we'll put those in the q1 OKR doc, and next week we'll clean that up, and then we have a few more weeks and we'll call it done
 - But here I want to talk about the bigger themes:
 - **Theme 1, story telling to Developer audience (both web fans and non-web fans):**
 - Announcing this work to the world
 - Dru: Planning to announce WPT dashboard and confluence at Google I/O (May-June) or Chrome Developer Summit (end of Sept or beginning of Oct)
 - I/O is more delicate - anything we do there has the full weight of google. Large smattering of developers. Things at I/O is not good for getting beta testers. But, it would be nice to have a story like: the web is more interoperable. This is a chance to get people who aren't already bought into the web.
 - [CDS](#): Our core users, already bought into the web and committed, and ready to try new things. This is a good place for beta users. This is the biggest chrome fan audience.
 - **Theme 2, things we need for stability:**
 - Running in > 24 hours
 - Automation
 - Bikeshed features

- Testing over changes in specs
- Presentation from Tab Atkins
 - How many of you have written spec? - 2
 - There are a lot of meta tasks around spec writing, including cross linking, etc. So Bikeshed is our tool to automate these tasks. I wrote it, starting 2-3 years ago. Its written in python. I wrote it as a response to wats (sp) Bert's C and awk based system and the respec system.
 - BS is a spec preprocessor that takes a markdown file, and makes a spec. It looks like this:



And then it spits out a spec:



- BS gives you other features
 - Table of contents
 - Issue tracking from github
 - Railroad diagram generation (styled like json.org)
 - Cross spec linking,

- which was the big missing feature from our former tools that cause me to write BS
- This is what lets us work on many related specs that depend on each other
- BS maintains a database of all the specs on the server in order to enable this
 - There is spider maintained by peter lins
 - It regularly crawls all known BS specs
 - Then there's 2-300 specs of definitions that you can link to
 - If you try to to link to something
 - The DB is here: <https://github.com/tabatkins/bikeshed-data/tree/master/data>
- There are some travis CI steps that will auto gen BS specs
- It is possible for a bikeshed build to break your build, and if that happens
- Philip, should al respec users use BS?
 - BS seems to be winning since respec doesn't have spec linking

2017-11-28

- Philip & Dru prompt: something awesome that happened since our last meeting
 - Jon - I think we're going to change testdriver.js to use the webdriver API directly and not selenium, and I think that will enable us to iterate on stuff fast, and not have a round trip, so I'm excited about that.
 - Mark - I'm not feeling victorious this morning. I learned how to do synchronization with go channels and that was fun
 - Robert - I finally started working on implementation for import notifications - and this week we'll be working on it and see who will opt in. The problem is that we'll be pretty close to the holiday, but anyways, this is coming.
 - Rick B - We sent Mozilla a cake to celebrate shipping quantum, and there was good buzz on twitter with Rah Rah we're working together to make the web better.
 - Boaz - interop bug with text rendering/overflow in inputs in Firefox is being fixed, a 6 year old bug finally being closed.
 - https://bugzilla.mozilla.org/show_bug.cgi?id=752790
 - <https://github.com/w3c/csswg-drafts/issues/1717>
 - <https://github.com/w3c/csswg-drafts/issues/1941>
 - The test: <https://github.com/w3c/web-platform-tests/pull/8080/files>
 - Sam - I have fewer issues assigned to me, which seems better, and I haven't closed them as wontfix \:D/
 - Dru - Many interesting discussions at TPAC, web-platform-test session was very well attended, which was very exciting. Dashboard metrics discussion was tense, but we learned a lot but we basically agree and just need to be careful about how we go about it.

- Mike - Getting the opportunity to work on wpt again, I'll be working on testharness.js to make it handle async cases better. This is something I had wanted to do for SWs, but didn't get the chance, and now with more demand, I'll be going to
- Rick W
 - Bocoup was accepted as a member of the MDN Product Advisory Board, will be attending first meeting on November 30th
 - Running wpt.fyi locally, working with Luke to refine updates and
 - Finished up my presentation for Numeric Separator Literal for TC39 meeting
https://docs.google.com/presentation/d/1E8yKRJwA4iX_EctpY48KGBwAsCtNZpTOz4wu7tbxTqE/
 - Learned a few things from Sam, he mentored me on a font test generation patch <https://github.com/w3c/web-platform-tests/pull/8396>
- Simon - I have been on a non google project, and it's been confusing, but I'm getting it working, and feeling good.
- Philip - I went to my first TPAC, and I think we had a lot of good discussions there, one on Tuesday, and a bigger one on Wednesday, and got good new priorities (e.g. getting mobile browsers running), and so that was exciting
- Dru
 - One of our 2018 OKRs that is the most poorly defined is to take the interop work we're doing and making it useful to developers
 - High level, our three related issues are
 - Improve interop
 - Pay down interop debt
 - For the interop bugs we know about, give developers tools for working around them
 - The goal I have for whatever we build is
 - Finding interop bugs is the most expensive part for developers: so a product for finding interop bugs before they hurt us
 - Whatever product we build, if it can provide data back to us to prioritize interop differences
 - I would like us to use the web platform data to create some kind of linting tool that looks for APIs that has failing WPTs, you get a warning with a link to the web platform test
 - Then once you see it, it would be cool if we had some kind of macro or comment language to say "I am working around the issue in this web platform test"
 - Then wptdashboard could pull this data in and show a star count for the number of codebases referencing x web platform test
 - Then when the bug gets fixed, your lint tool gives you a new warning as a developer saying this work around is no longer needed
- Jon

- Right now WPT is geared towards browser engineers, and we run this against the latest browsers, the information bleeding edge
- Dru:
 - Good point, maybe there could be a config file for “browsers I’m targeting”
- Mark:
 - There’s an established syntax for this in the node community, and we should use that <https://github.com/ai/browserslist>
- Dru:
 - Does this feel like it’s solving the correct problem?
 - Jon: the other problem I was thinking about is web developers not using vanilla js, and I think that mapping your doing is not as easy as your thinking - e.g. do we need to create further mappings for each popular framework?
 - Boaz: also issues with mapping API usage to WPT repo
 - Boaz: also issues with conflict with the goals of not pointing developers to look at interop
 - Dru:
 - The goal wouldn’t be to direct developers to the dashboard
- Mark
 - In the vein of what Jon was saying about solving the right problem, I think it will be important to get more data from web developers in the form of surveys
 - Maybe we should provide a tool for framework developers to provide this data
 - Dru:
 - A lot of people use frameworks as a compat shim
 - You could imagine going in a totally different direction to identify the 20 top interop issues and building a compat lib
 - Boaz
 - Checkout modernizr
- Jon
 - Failing web platform test doesn’t mean that there’s an interop bug they care about
- Boaz
 - Also worth checking out the way Babel targets different environments, we’ve talked about
- Simon
 - I remember discussing this at TPAC, and one of the ideas is being able to annotate tests in WPT that have known web compat issues

2017-10-31

Around the room - victory edition:

- Dru - excited about potential developer tools around interop.

- Robert - Successfully figured out what's going wrong with rebaseline tool. Draft CL ready.
- Luke - Pushed a few merges. Endpoint that serves the latest json results for wpt.fyi. Can also look at an arbitrary run as well.
- Rick - Finally landed changes for UMA backend that serves Chromestatus for usage data.
- Jon - Playing around with WPT dashboard and Chromedriver.
- Mike - TPAC happening next week! Looking forward to what comes out of it.
- Philip - Transitioned whatwg.org to new hosting

Presentation - Luke on Blogger interop issues

- Writing new templates for Blogger publishing
- Deciding what browsers to support for new templates - decided largely by market share (small team, wanted to hit the most users with the amount of work they could do)
 - A few browsers that had high usage in areas that were strategically important
- Transparent is technically rgba(0,0,0,0). Most browsers adjust for this when you have a gradient to transparent, Safari does not.
- Browsers don't agree on line height 'normal' value
- Image resizing different across browsers
- Scrollbar included in the viewport for some browsers, but not all
- "Copy to clipboard" only works in some browsers and on some platforms
 - Took about a week of time to get this working on all browsers
- Some features dropped entirely when they don't work cross browser

2017-10-24

Dru: set up an automatic rotation for weekly talks, this week it's Jon's turn.

Around the room, this week let's go with a challenge:

- Dru: (predictability) thinking about Chrome-only Google properties. A challenge is that Google internally doesn't have best practices. No lever to pull, but we need to move it.
- Philip: Working on code for find specs not in <https://foolip.github.io/day-to-day/> and finding that a lot of the data needs to be ignored.
- Raphael: working on "test as you commit" policy for devices WG, lots of overhead and back and forth. Also sent PR for [Gamepad](#) and Gamepad extensions.
- Matt: Welcome! New from Bocoup, working on getting started with everything.
- Quinten: thinking about future ownership of [Blink>Infra](#). (Philip comment in doc: there's an experimental part of the ecosystem infra rotation for it.)
- Lorin: making sure that Matt has everything he needs to roll into Bob's place. Also started looking at OKRs for Bocoup itself.
- Rick: web properties that don't behave the same in all Chromium-based browsers. We also don't have a consistent view of interop bugs, arguing about whether certain bugs belong in crbug.com or not. Personally want to say that any time Chrome ships behavior

that isn't standardized and covered by a shared test suite, that's a Chrome bug, our problem.

- Mark: No hermetic build environment for wpt.fyi. Working on protobuf instead of JSON, getting the build dependencies (protoc etc.) in there.
- Jon: [WebRTC running issues](#) with wpt run - everything seems like it should work but still hitting issues.
- Robert: Just getting back from vacation and trying to get caught up on missed work. Trying to figure out [what's wrong with rebaseline-cl](#).
- Luke: Working on an endpoint to access the latest run data for a given browser. Will enable us to put the expected results in PR bot as latest run.
- Tab: Mapping tests back to specs and trying to find what areas aren't tested well. Will send design doc to ecosystem-infra.

Rick: predictability and wpt was called out at CDS.

Presentation (Jon): Metric Brainstorming

- 3 groups: Browser vendors, web devs, and internet users
- Current metrics: WPT pass rate, WPT adoption rate, minimizing time to ship APIs across browsers
- Not directly measuring impact to developers (only correlations)
- Other areas we've looked at
 - Web developer survey
 - Twitter sentiment analysis
- Could frameworks be a more concrete way to gauge interop health?
 - Analyze interop code of Angular perhaps?

2017-10-16

Meeting moved an hour because of office meeting in Waterloo.

Doing the rounds, something interesting in the last week?

- Luke: blanking.
- Rick: listened to Philip's Web Engines Hackfest talk, some discussion at end.
- Jon: Trying to figure out where people are at with standardizing Permissions in WebDriver. Seems like some people want to wait for TPAC. Brian Burg has already implemented something in SafariDriver.
- Mark: trying to compile and use protobufs in different languages. Context is thinking about stats for the dashboard.
- Lorin: Matt Lee is starting this week, will work on ecosystem infra stuff.
- Mike: Same as Jon, the Permissions WebDriver extension. A bit surprised that the WebDriver folks want to take in the main WebDriver spec, seems better strategically to

promote the pattern of extensions. Wondering if it could get in the way of REC. A surprise that something suddenly appeared in SafariDriver.

- Philip: Test262 governance / 2-way sync meeting. V8 team wants to do 2-way sync like WPT, but test262 already in better shape and has a stronger code review culture than WPT. Looked into what is actually required (enforced by lint) and found it doesn't actually capture what people are concerned about. Interesting and unclear what the outcome will be.
 - Mike: there's also <https://github.com/w3c/web-platform-tests/pull/7855>

Presentation/discussion on [Ecosystem Infra Stats](#)

- Which of these stats are already solid?
 - WPT commits
 - Added servo exports
 - Chromium exports not going up and up, more of a cycle
 - This seems useful
 - Import/export metrics
 - Robert has defined a symmetric metrics for import/export latency in Q4, these will make sense in the long term as a metric we can't allow to regress, but which should require less and less of our time to achieve.
 - WPT usage in chromium
 - Been going up and down a bit around 20%, not moving in the direction it needs to be.
 - Monitoring of web properties:
 - WPT pass rates
 - Don't yet have month-to-month yet
- Rick: metrics are really important, especially for this team
 - Need to try to quantify our impact - even though it's only a rough correlation with what we're really trying to achieve
 - Need both Leading metrics vs. lagging metrics.
- Which are problematic? "Usage in Chromium" is problematic.
- What additional stats do we need for the long term?
 - Something that tracks web developer productivity/satisfaction.
 - Mark: there's also the question of interacting directly with the web platform vs. using frameworks.

2017-10-10

This week's question: recent triumph or challenge.

- Philip: [Comparing WPT contributions from Chromium between 2016 -> 2017](#). Huge improvements from 2-way sync.
- Rick: Full source level debugging of Google web properties on Firefox, finally found a way to do it! Made compat analysis much easier.

- Mark: Rewrote object graph API extraction. More consolidated process and lots of debug metadata.
- Jon: Spec compliant flag in ChromeDriver will now enforce what you can/can't use.
- Luke: Got a staging environment for wpt Travis setup working.
- Bob: Triaged all the infra issues in wpt, added prio label to all of them.
- Sam: Debugging testharness.js pr. JS debugging horrible as always, WebDriver command [execute_async_script](#).
- Lorin: Onboarding Bob's backfill going well. Good direction!
- Mike: [PR for integrating WebDriver and permissions API](#) getting feedback.
- Raphael: Was able to reach out to the [Devices and Sensors WG](#) and made great progress. Would cover 11-13 specs with testing policy.
- Tab: Rewriting the HTML parser for Bikeshed.
- Dru: Something to do with this week's presentation.

Dru presenting the wpt.fyi interop stats PRD (product requirements document).

-

2017-09-26

- It's OKR planning time! [Q4 OKRs](#)
 - WPT import/export tool
 - P0 is to maintain the latency quality bar we set in Q3
 - Reduce pain points that Blink developers have with WPT
 - Data infra
 - Close any urgent bugs for confluence after launch
 - Publish a few representative, high value queries for WPT results
 - Chromedriver
 - Increase automation of WPT
 - Engineering discipline for Chromedriver
 - Better metrics
 - Enable WebRTC
 - Test quality
 - Regression/review tool
- [everyone] Finalize OKR drafts for your work by Friday:
 - All KRs listed
 - Scoring
 - Relative priority between your O's
- [foolip + dknox] global OKR review before next week's meeting

2017-09-14

- No meeting next week due to BlinkOn
- P0 goal brainstorming

- Mark:
 - (1) Drive adoption of opt-in infra (e.g., Confluence, WPT instead of non-WPT LayoutTests), OR
 - (2) Decrease opt-in-ability of these tools (what would it look like for Confluence to be "enforced" the way a precommit hook is?)
 - P0 is for Blink engineers. Also important long term is driving adoption by other browser engineers.
- Bob:
 - Make web-platform-test infra "better". Ways to measure "better": could be the timeliness of addressing priority:urgent infra issues (within 24 hours), addressing priority:roadmap issues in a reasonable time.
- Sam:
 - Get rid of the CSS build system. There are a bunch of random-looking requirements on CSS tests, that are onerous for test writers.
- Quinten:
 - Goal: WPT is a conformance and regression test for all of web platform.
 - Goal: WPT is high-quality, useful and has "good coverage".
 - KR: NN% Coverage of X spec is covered by tests.
 - KR: It's easy to see the corresponding tests for each spec section.
- Philip:
 - We want WPT to be the main test suite that all vendors use
 - Want binary answer: for feature X is WPT the main test suite in use for each engine
 - Want to drive this to 100%
 - Means quality will follow, have to upstream all your tests
- Rick:
 - Deliver on the original vision, to have basic software engineering discipline for the web platform as a product. In 2018, should stop the bleeding, if not necessarily solve the backlog. Make wpt/test262 a requirement for shipping in Blink.
 - Harden spec policy to have tests any time there's an implementation.
- Robert:
 - Make chromium developers use WPT more, make them care more about status
 - Eg. use % of notifications opt-in as metric - how many teams choose to be notified of new failures in their areas
- Jon:
 - O: Automate manual tests in WPT (P0)

KR: Almost all new tests are automated

Owner: kereliuk

Scoring:

- Let T be a date
- Let N be the number of new tests written between Sept 1st and T
- Let M be the number of manual tests written between Sept 1st and T
- $\text{Score} = \max\{0, (1 - M/N * 10)\}$

KR: Sort existing manual tests and plan for automating them/outsourcing

Owner: kereliuk

Scoring:

- Start at score 1
- Let T be a date
- -0.01 for every manual test without a plan for automation at the end of T
- -0.001 for every manual test at the end of T

Currently (Sept 14th) `find . -name "*manual*" | wc -l` is 1538

- Raphael
 - Change in culture to upstream-first approach to testing. Write WPT tests by default, upstream as many as possible.
 - How to we accelerate upstreaming existing tests? Culture change? Teams feel responsible for the suite for their features.
 - Make chromium teams more aware of WPT failures, failure rates and spec changes
 - How spec editors notify implementers varies a lot by spec. Some spec maintainers file bugs for implementers, but most not.
- See also [wpt.fyi use cases from James Graham](#)

2017-09-05

- Main focus areas
 - Robert: WPT import/export tools
 - John: Chromedriver
 - Mark: Interoperability insights - what browsers are shipping what APIs?
 - Bob: WPT running automation (dashboard and PRs)
 - Sam: Automating manual WPT (specifically things that need a click)
 - Leon: Working on open source contributions from Intel (in Beijing)
 - Mike: Testing for webdriver and service worker
 - Raphael: Working on ecosystem infra rotation, bindings, webassembly

- Dru: PM for all of the predictability efforts, alongside Rick. For ecosystem infra, big focus is 2018 planning. User weighted values for interop tests.
- Quinten: WPT importer. Presentation on WPT sync info at BlinkOn
- Philip: TL for ecosystem infra. Q4, 2018 planning.
- Slides: [Ecosystem infra... and culture?](#)
- How can ecosystem infra affect the culture of web standards/platform engineering to be more interoperable?
 - Culture is behavior - people wanting, or being willing, to invest in interoperability
 - Vs. infrastructure - the tooling that enables individuals to invest in interop effectively
 - A lot of our current culture work falls into two categories:
 - Understanding why people don't want to use WPT right now
 - Incorporating WPT into our launch processes
 - Culture and infrastructure will affect each other. Need to balance where we invest.
 - Case study: Getting specs to test normative changes
 - Initial goal: 5 specs in Q2, 10 in Q3, adopt a policy of using WPT to test normative changes
 - Results: 15 in Q2, 43 in Q3
 - Why did this work?
 - Start where we already have credibility
 - Took skepticism very seriously
 - Focus on low-hanging fruit and easy steps to get momentum rolling
 - Start with private conversations before proposing things in large public forums
 - Keep a list of success cases to justify future asks
 - Look for allies who can help you argue your case
 - Things to avoid:
 - Trying to do things by sheer effort, without infra in place to support
 - Asking for people to invest before there's real benefit for them
 - Brainstorming for the future
 - Are we balancing infra vs. culture correctly?
 - Underestimated the value of iterating on culture while we develop the WPT infra
 -

2017-08-29

This week's snippets thread on [ecosystem-infra](#).

Going around the room, talking about a current challenge or something frustrating.

- Philip: Fullscreen regression that are hard to bisect and Android WebView. Doesn't build!

- Jon: looked at Dru's repo and instructions from last week, found it difficult to add a new metric to it.
- Mark: 2 threads are better than 1, have been trying to get AppEngine to quickly serve data.
- Robert: Fixed data loss issues last week, one of the fix had an unexpected side effect. Now under control. Good news: if the importer or exporter is red, that means something is actually wrong. Jon: the rotation was super easy today, took about 6 minutes.
- Quinten: Red and green haven't corresponded very well to success or failure before, now it's better than ever before. Spent a lot of time rebaselining to get rid of the trailing space in -expected.txt files.
- Rick: frustrated that I haven't been doing as much hands-on work as I want to, instead working with Dimitri on strategy. This team is a very strategic team. Always think about how we're having an impact on the web.
- Dru: Tried reaching out for Firefox, Edge and Safari for wpt pain points, and it's been hard to get much engagement. Some discussion, but no reply from Apple yet. Microsoft thinks a bunch of manual tests could just be converted to automated tests. Seen "daily".

Rick: we had a meeting with plh@w3.org about web-platform-tests. Floated the idea of not supporting charter renewals where testing isn't taken seriously. Once there are no good reasons for using wpt, we should probably do that.

Jon presents! Slides here.

2017-08-22

This week's snippets thread is on [ecosystem-infra](#).

Repeating snippets as status updates not great, so trying to instead speak about one current challenge.

- Challenges from the week
 - Correlating interop bugs to user data
 - How can we find and report interop bugs more cheaply?
 - The ecosystem-infra rotation is more time consuming than ideal
 - But also rewarding, many small things to fix
 - Recommend that if you find bugs with import/export, just file a bug and cc Robert, Jeff, and Quinten to follow up
 - Trying to come up with a better serialization format for API confluence (87MB currently)
 - Should we try to cache on the client or just download every visit?
 - Chromedriver has been very neglected and isn't spec compliant. How do we improve spec compliance while balancing back compat?
 - Sounds like a good presentation for next week!

- Keeping import/export stable still time consuming
- Permissions API is not being responsive
 - ~~AI foolip: ping Googler spec editors to try and get a response.~~ Done in email, folks in bcc.
- [API confluence research](#)
 - How would we set OKRs around these numbers?
 - Trying to drive down the time to a second implementation
 - Could we time-weight the “successful first ship” numbers?
 - This might make it hard to ship big, ambitious features
 - Could we weight apis by usage and aim to ship first on APIs with high usage?
 - May miss enterprise or other biases in UMA

2017-08-15

This week we did [Q3 OKR](#) mid-quarter check-in and nothing else, going through the list of OKRs and saying something about the scoring.

2017-08-08

- Mid quarter OKR check-in next week
- Interesting things this week and next
 - Digging into the data we’ve generated for API confluence
 - Compat hackathon with Moz, Bocoup, Search
 - A bunch of outdated WPT from Selenium, want to explore whether we can import them
 - Working on interop principles doc (mirror to [compat principles doc](#))
 - Pushing browser versioning data for API confluence
 - Lots of bug fixes for API confluence
 - Removed getStreamById from RTCPeerConnection
 - % of Chromium commits that used WPT was down in July, need to understand why
 - PR results dashboard deployed, need to merge one PR in WPT to use it
 - CSS F2F last week in Paris, got some agreement to require tests for normative changes after CR
 - Concern about workload of writing tests for everything
 - Reviews for tests being a bottleneck
 - [Blog post](#) on unspecified components of a spec published
 - UI improvements and running script landed for [wpt.fyi](#)
 - Working on getting the “interoperable subset” of the web - 37% of WPT pass in all browsers. [script here](#).
 - Working to reduce Chromium’s flakiness when running WPT
 - Verifying data for IDL diff tool

- API confluence demo
 - Catalog shows tactical changes that can be made
 - Aggregate metrics will help make decisions at leadership level

2017-08-01

- Weekly snippets, plan to do it via email reply chain starting next week.
- Setting up a proper Ecosystem Infra Rotation.
- Jon:
 - working on the actions API for ChromeDriver, currently on the DevTools bits.
 - There are ChromeDriver tests that come from Selenium, which is a similar situation from web-platform-tests.

The rest of the meeting turned into a wpt sync meeting:

Robert: starting with some easy bugs in import/export, like the pagination of GitHub requests needed to get all PRs with the chromium-export label. Continuing to work on easy bugs, like suppressing patch failures from logs.

Quinten: Importer is failing today. Last week worked on starting CQ bots at first upload, and other small things. Also worked on getting the importer to reapply exportable changes to not be blocked on export. Curiously haven't seen it actually happen yet.

Philip: Seems like exports are getting stuck a lot less, a pleasant surprise. James Graham has done work (building on the wpt command) to not run some Travis jobs that aren't needed, which may have contributed to it.

Jeff: The "do not merge yet" labels are now automatically removed. Instead have been working on finishing up some work on the dashboard. Hope to still keep working 20% on it when Bob starts working as well.

Philip: Import stats are going in the right direction. Jeff has updated the export stats, they are looking good, matching our sense that exports are getting stuck less.

2017-07-25

- [Web compat hackathon in WAT](#) - Aug 10th
- WPT dashboard
 - WebAssembly working on interop testing, chose to use WPT instead of elsewhere in order to get WPT dashboard, import/export tooling
 - Domenic (HTML editor) really excited by WPT dashboard
- Updates
 - Mark

- Web IDL diff reviews, couple weeks left for Michael's internship
 - Confluence: planning for code freeze Sept 1, demo at BlinkOn. Working to close a bug a day. Should be achievable
- Rick
 - Working on planning compat hackathon
 - Ecosystem infra rotation - only one issue - importer failures Quinten is investigating. Confused by log error, assigned bug to Robert.
- Robert
 - Ramping up on WPT importer. Overhauled W3CImportExpectations to remove OWNERS redundancy.
 - Now getting familiar with exporter.
- Quinten
 - Working on fixing import failures - think it's a recently introduced bug
 - Imports should no longer be blocked on exports - should continue without clobbering it.
 - Fixed bug with CQ flakiness causing aborts
- Dru
 - Preparing to visit WAT in 2 weeks, compat hackathon
 - Want to try to synthesize data from various tools, what story they tell
- Bob
 - Reviewing James's big test running infrastructure changes
 - Working on landing PR results app, getting reviews from James
 - Have servers setup, some security work remaining
- Jeff
 - Fixed an exporter bug where occasionally CL wouldn't get landed.
 - Dashboard
 - Landed test file results pages, arbitrary navigation
 - Have Edge results
- John
 - Working on pointer actions API for chromedriver, working with devtools team to dispatch actions
 - Also working on chromedriver bugs
- WPT dashboard demo from jeffcarp
- Snippet chain
 - Dru will start a thread on ecosystem-infra weekly for everyone to share status publicly

2017-07-18

- PR buildbot demo from bob@bocoup
- Status updates

2017-06-27

Agenda

- Q2 okr review
 - Automation
 - Input event automation
 - More design work than expected
 - Have some code landed but not as far along as originally planned
 - One other event can be automated
 - Import/Export
 - Frequency of imports is not much higher than before
 - Focusing more on the failure cases (e.g. accidental reverts)
 - Want to focus on making imports require less manual intervention
 - Export PR tool seeing increasing use
 - All of Chromium will switch to using gerrit soon, will be able to build tooling (maybe specific to gerrit) to block CLs from landing until upstream PR is green
 - Dashboard
 - Prototype from Q1, iterating on it throughout Q2
 - Results on the dashboard
 - Almost daily Chrome and Firefox running on VM, haven't gotten to Edge or Safari yet though
 - Showing test results for WPT PRs
 - 0.85 for reliability of tests running
 - Should set more aggressive goals next time
 - SW test uploading
 - A few tests remaining
 - Remaining tests are still discussing edge cases
 - 0.91
 - API confluence / IDL diff
 - Modest progress this quarter
 - Biggest work for IDL diff was porting work from proof of concept to a more productionised repo
 - Dramatically overshoot our goal around closing IDL diff bugs
 - For API confluence, lots of infra improvements to improve performance.
 - UI polish bugs remain
 - A couple performance issues
 - Have a draft launch bug, need to do security review
- Q3 brainstorm
 - All: Dru + Philip + ?Rick
 - WPT automation
 - Sam

- Jon
 - Mike
- Dashboard
 - Jeff + someone from Bocoup
- Import/export
 - Jeff + Quinten
- API confluence + IDL diff
 - Mark
 - Michael
- Upstream WPT infra
 - Bocoup folks (based on who's working on it currently)
 - Bob
 - Jeff
 - How do we integrate Gerrit with the upstream infra?
- Testing standards culture
 - Philip
- What does it look like to affect culture?
 - Should make this more concrete
 -

2017-06-20

- Welcome and introductions, we haven't all met before.
 - Present: Mark, Rick, Bob, Dru, Philip, Sam, Jeff, Mike, Michael
- Q2 OKR scoring
 - Need to score by end of month
 - [AI dknox] send out list of OKRs for ecosystem infra team and who owns each
- Q3 OKR planning
 - Should share ecosystem infra charter with team
 - [AI foolip] to share charter with platform-predictability
 - Vision:
 - Web platform is actually a collection of implementations
 - Don't have the engineering discipline / infrastructure to build the web platform as a cohesive product
 - Currently, no one is surprised when browsers aren't interoperable
 - Ultimately, we want the web platform to behave as one entity by default
 - OKR areas
 - WPT is increasingly used in browser/standards work
 - Import / export
 - % of blink engineers writing WPT with their code
 - Dashboard
 - CI for github
 - Automation story

- Automatically detecting differences between browsers
 - API confluence
 - IDL tool
 - Improve engineering discipline in standards work
 - WPT usage - overall commit count in WPT
 - # of specs adopting a WPT policy
 - [Al dknox + foolip] solidify OKR groups and schedule follow up meetings to set OKRs by next week
- Naming
 - No strong opposition to ecosystem infra
 - Infra is a bit broader scope than tooling
- Things we want to cover in the meeting
 - Around the room updates
 - Small presentations on the status of our work