About me: software engineer at Google, currently in Seattle, working on Gerrit for 7 years. Google runs hosted Gerrit, but we don’t fork Gerrit. (We do remove some features, e.g. SSH) Some of this work is done by Google, some from others in the community, but this is all stuff we want to make available to everyone.

This is not just a list of new features. I’ve done that talk, it’s useful, David Pursehouse told me he assumed that’s what I was doing, but I wanted to mix it up. I want to hit the highlights and show some trends, to hopefully convince you that upgrading is something you want to do.
Out with the old, in with the new
Before we get into Gerrit itself, let's take a look at the website.
Gitiles is nice. Live updating is nice. But this uses custom parser code based on CommonMark, and it's really hard to add new features, to say nothing of custom themes.
Unfortunately this isn't live rendered, but this means we can use a system with a lot more features (Jekyll).

Renders clean static HTML, served from a simple static hosting service (FireBase Hosting).

Additional features it's not feasible to build into Gitiles:

* Client-side search
* Dropdowns
* Navbars
* Wide variety of well-designed templates.

It's 2018, so we can set up triggers to automatically build and deploy the Jekyll site whenever master is updated (cough, Luca).

Thank you Dave Shevitz, Google Tech Writer. No longer primarily assigned to Gerrit, but since he left the team he already completely redesigned the website template and rendering like, last week.
PolyGerrit

Sorry, these slides are from 2.16-rc3; I finished them literally minutes before 2.16 final was released.
This looks ancient to me but I'm guessing most people here are most familiar with this interface. Personally, I don't have a problem with it. But we've learned that FE devs and UX designers will not touch GWT. Plus, GWT the project won't last forever.
Still some rough edges here, e.g. dropdown style inconsistency
In 2.15, change screen looks pretty good but there are still a chunk of missing features. The recommendation is, "if you can't do X in PG, switch back to GWT".
Notable changes:
* Material Design
* Redesigned submit requirements (label status)
* Adjust whitespace/layout (which is a constant battle)
* PolyGerrit -> Gerrit
* Not pictured: inline edit with CodeMirror plugin
* Update Change button

Instead of just porting features over, this one has had involvement with our UX design team (arnabb@)

Plus...
Thanks Kasper (from Lund hackathon)
Changes in 3.0 are more modest. Some UI tweaks but not as many. 3.0 is more about GWT parity in the long tail of features (e.g. plugin provided settings).
### Outgoing reviews

<table>
<thead>
<tr>
<th>Title</th>
<th>Owner</th>
<th>Project</th>
<th>Branch</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer guide: Update command to invoke google-java-format</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Upgrade goavis to 2.0.0-rc</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Upgrade protobuf-java to 3.8.1</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Add Even Stenberg to maintainer</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Bazel: Require minimum bazel version 0.17.1</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>bazelits: Replace native git repository with skyark rule</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Upgrade bazelits to latest revision on master</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Add note about aligning upgrades of pretty with plugzgit6es</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Upgrade JGit to 5.1.2.201610061102-r</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Update URL for maven version rules description</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Add Sub-NetVar Support To Griffies</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Require minimum Bazel version 0.19.0</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Bazel: Consume versions directly from lib-versions.bzl in skylib</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Fix Truth dependencies</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Fix missing reference name in JSON output</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>FakeItResponse: Explicitly specify charset in PrintWifiResponse</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Increase severity of DefaultChannel to ERROR</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Upgrade JGit to 5.1.3.2016102000300-r</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
<tr>
<td>Format all Java files with google-java-format</td>
<td>Administrator</td>
<td>griffies</td>
<td>master</td>
<td>9:13 PM</td>
</tr>
</tbody>
</table>

### Incoming reviews

(None)

### Recently closed

(None)
# Outgoing reviews

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Status</th>
<th>Owner</th>
<th>Assignee</th>
<th>Project</th>
<th>Branch</th>
<th>Updated</th>
<th>Size</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format all Java files with google-java-format</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+3</td>
<td>-6</td>
</tr>
<tr>
<td>Upgrade JUnit to 5.1.3.2018020035000 r</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+5</td>
<td>-5</td>
</tr>
<tr>
<td>Increase severity of DefaultCharset to ERROR</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+1</td>
<td>-1</td>
</tr>
<tr>
<td>FailHttpStatusResponse: Explicitly specify charset in PrintWriter</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+7</td>
<td>-2</td>
</tr>
<tr>
<td>Fix missing reference name in JSON output</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+13</td>
<td>-1</td>
</tr>
<tr>
<td>Fix Truth-dependencies</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+3</td>
<td>-3</td>
</tr>
<tr>
<td>Bazel: Consume versions directly from lib-versions.bzl in skylark</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+1</td>
<td>-1</td>
</tr>
<tr>
<td>Require minimum Bazel version 0.19.0</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+2</td>
<td>-2</td>
</tr>
<tr>
<td>Add Sub-Netizer Support To Gitiles</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+63</td>
<td>-1</td>
</tr>
<tr>
<td>Update URL for maven version rules description</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+1</td>
<td>-1</td>
</tr>
<tr>
<td>Upgrade JUnit to 5.1.2.2018061102-r</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+5</td>
<td>-5</td>
</tr>
<tr>
<td>Add note about aligning upgrades of prettier with plugins/gitiles</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+1</td>
<td>-1</td>
</tr>
<tr>
<td>Upgrade bazel to latest revision-on-master</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+3</td>
<td>-1</td>
</tr>
<tr>
<td>bazeltest: Replace native.gitRepository with stylerisk rule</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+3</td>
<td>-1</td>
</tr>
<tr>
<td>Bazel: Require minimum bazel version 0.17.1</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+13</td>
<td>-8</td>
</tr>
<tr>
<td>Add Even Steinberg to .mailmap</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+2</td>
<td>-1</td>
</tr>
<tr>
<td>Upgrade protobuf-java to 3.6.1</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+2</td>
<td>-2</td>
</tr>
<tr>
<td>Upgrade-guava to 26.0-py</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+2</td>
<td>-2</td>
</tr>
<tr>
<td>Developer guide: Update command to invoke google-java-format</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+1</td>
<td>-1</td>
</tr>
<tr>
<td>chore: revise developer guide</td>
<td>administrator</td>
<td></td>
<td>gitiles</td>
<td>master</td>
<td></td>
<td></td>
<td>9:13 PM</td>
<td>+2</td>
<td>-2</td>
</tr>
</tbody>
</table>

# Incoming reviews

No changes

# Recent closed

Gerrit 2.16, 3.0, and beyond / Gerrit User Summit 2018
<table>
<thead>
<tr>
<th>Subject</th>
<th>Status</th>
<th>Owner</th>
<th>Assignee</th>
<th>Repo</th>
<th>Branch</th>
<th>Updated</th>
<th>Size</th>
<th>CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Format all Java files with google-java-format</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>XS</td>
<td></td>
</tr>
<tr>
<td>Upgrade JRe to 5.1.3-2018100395-r</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Increase severity of DefaultCharset to ERROR</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>XS</td>
<td></td>
</tr>
<tr>
<td>FakeHttpServletResponse: Explicitly specify charset in PrintWriter</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>XS</td>
<td></td>
</tr>
<tr>
<td>Fix missing reference name in JSON output</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Fix Truth dependencies</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Bazel: Consume versions directly from lib/versions.bzl in skylib</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>XS</td>
<td></td>
</tr>
<tr>
<td>Require minimum Bazel version 0.19.0</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>XS</td>
<td></td>
</tr>
<tr>
<td>Add Sub-Notification Support To GItiles</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>Update URL for maven version rules description</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>XS</td>
<td></td>
</tr>
<tr>
<td>Upgrade JRe to 5.1.2-20181001102-r</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Add note about aligning upgrades of prettify with plugins/gitless</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>XS</td>
<td></td>
</tr>
<tr>
<td>Upgrade bazelci to latest revision on master</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>XS</td>
<td></td>
</tr>
<tr>
<td>bazelci: Replace native git repository with skylark rule</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>XS</td>
<td></td>
</tr>
<tr>
<td>Bazel: Require minimum bazel version 0.17.1</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>Add:Euri Stemberg lr .mailmap</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>XS</td>
<td></td>
</tr>
<tr>
<td>Upgrade protobuf/java to 3.6.1</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>XS</td>
<td></td>
</tr>
<tr>
<td>Upgrade guava to 28.0.0</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>XS</td>
<td></td>
</tr>
<tr>
<td>Developer guide: Update command to invoke google-java-format</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>XS</td>
<td></td>
</tr>
<tr>
<td>chore: revise developer guide</td>
<td>-</td>
<td>Administrator</td>
<td>-</td>
<td>gitless</td>
<td>master</td>
<td>9/13 PM</td>
<td>XS</td>
<td></td>
</tr>
</tbody>
</table>
So yeah, 3.0 is pixel-identical to 2.16. Notice a trend: less chrome surrounding the dashboard, so your content is more front and center. This allows us to add whitespace to make things slightly less cramped without actually changing the amount of content on any one screen.

This is the dashboard for an obviously experienced user. But what about inexperienced users?
Actually, something did change!
### Gerrit 2.16, 3.0, and beyond / Gerrit User Summit 2018

**Outgoing reviews**

- **Push your first change for code review**
  
  Pushing a change for review is easy, but a little different from other git code review tools. Click on the 'Create Change' button and follow the step by step instructions.

- **CREATE CHANGE**

**Incoming reviews**

- **Format all Java files with google-java-format**

- **CCed on**
  
  - No changes

- **Recently closed**
  
  - No changes

---

**Create change**

<table>
<thead>
<tr>
<th>Repository</th>
<th>Branch</th>
</tr>
</thead>
<tbody>
<tr>
<td>gfiles</td>
<td>master</td>
</tr>
</tbody>
</table>

- If you haven't done so, you will need to clone the repository.

---

Powered by [Gerrit Code Review](http://review.example.com) (2.16-e3-165-gce730be16)

Gerrit 2.16, 3.0, and beyond / Gerrit User Summit 2018
Push your first change for code review

Pushing a change for review is easy, but a bit of extra effort is needed to use the change review tools. Click on the "Create Change" button to get started.

Create change commands

1. Make the changes to the files on your machine

   $ git commit

2. If you are making a new commit use

   $ git commit --amend

   Or to amend an existing commit use

   $ git commit --amend

   Please make sure you add a commit message as it becomes the description for your change.

3. Push the change for code review

   $ git push origin HEAD:refs/for/master

4. Close this dialog and you should be able to see your recently created change in "Outgoing changes" section on "Your changes" page.

   DONE
NoteDb
### NoteDb

**Simplicity**
- All data stored in one place (git)
- No more database management

**Consistency**
- Replication natively includes code review data
- Atomic backups are as easy as "git clone --mirror"

**Auditability**
- No mutable records; every modification is a git change
- Every action is logged.

**Extensibility**
- Plugins can add new fields
- ...eventually

**New Features (really!)**
- Federation
- Interoperation

---

I continue to copy and paste this slide from older presentations. But now it's real! Really real!

Also since last time: one of our partners is doing actual federation, copying review data from a googlesource.com site to their own site (with more plugins)

Go project has a tool called maintnr that bypasses the whole REST API to use an in-memory copy of NoteDb data

The best part is I didn't know about either of these products a month ago. People are using this tool in ways I hadn't envisioned.
$ git log --no-decorate refs/changes/12/12/meta
commit 7c614f3cbf8e5e37b195bcf824828e1d24815f39
Author: Administrator <1000000@321034a2-38e6-4248-9d0d-32d2f31e68c6>
Date:   Wed Nov 14 21:13:23 2018 -0800

Create change

Uploaded patch set 1.

Patch-set: 1
Change-id: I2013bf5798a098f0340e3cf6a0ee91286432edc0
Subject: Add Sub Navbar Support To Gitiles
Branch: refs/heads/master
Status: new
Topic:
Commit: e8f7411f27fe0dacf8096b0fe0c1259f87e402c85
Tag: autogenerated:gerrit:newPatchSet
Groups: 966c93a524cdb445545b6f50be8e1633759966c1
Private: false
Work-in-progress: false
NoteDb happened

Production ready in 2.15; Google has been using in production for ~1.5 years.
Online or offline migration in 2.16; still stores a tiny bit of data in ReviewDb
3.0 will not know how to talk to a database.
Submit Rules
Show of hands: who **enjoys** writing Prolog rules? Or better yet, debugging them? And that's assuming you aren't writing your own predicates.

I can squint at this and sort of figure out what it does: if there are no unresolved comments, the "All-Comments-Resolved" label is ok, if it's >0, then it's needed. But if you asked me to write this from scratch...
Java!

```java
public class NoUnresolvedCommentsRule implements SubmitRule {
    private static final SubmitRequirement REQUIREMENT =
        SubmitRequirement.builder()
            .setField("unresolved_comments")
            .setFallbackText("Resolve all comments")
            .build();

    @Override
    public Collection<SubmitRecord> evaluate(ChangeData cd, SubmitRuleOptions options) {
        Integer unresolvedComments = cd.unresolvedCommentCount();
        SubmitRecord sr = new SubmitRecord();
        sr.requirements = Collections.singletonList(REQUIREMENT);
        sr.status = unresolvedComments == null || unresolvedComments > 0
            ? SubmitRecord.Status.NOT_READY
            : SubmitRecord.Status_OK;
        return ImmutableList.of(sr);
    }
}
```

Ok, sorry, this is Java so it's a little verbose. But the logic is transparent!
And with scripting plugins these can be even simpler, written in Groovy (as Luca reminded me yesterday).
Don't want to steal Han-Wen's thunder, he will talk much more about this in his talk.
(2.16)
Logging
This is more exciting to developers like me who can never remember how to use the slf4j API.

Huge thanks to Edwin for all the migration work, including adding some new features to the Flogger project
I don't

(2.16)
Ok, that wasn't very exciting. What is more exciting is the new features Flogger enables.
A user files bug #123 complaining about latency of reviewer auto-completion. You can get more information than you would ever want about the latency!
Just search for "bug/123" in your logs!
Because of this feature we are able to add **much** more detailed logging, which is there only when you want it.
* Every time a plugin is invoked
* Every time a metric is recorded
* Every index query
* etc. It's...possible we now have too much logging, but honestly this is a good problem to have.

Notice that we didn't have to mention tracing at all in the log lines.
Flogger lets you set a per-request logging context which is implicitly logged.
Git Protocol v2

Google's git core team in Sunnyvale drove this work.
Thanks Brandon Williams (core; now at Facebook), Jonathan Tan (JGit)
Historically, the git protocol hasn't been very extensible (long story; like all good C stories, it involves null bytes), so this has made it hard to add new features. What we have today is a few features to entice you to switch, but there are more in store.

You read that right: Chromium has almost 1M refs, and you have to fetch them all just to get the value of "master"
The 124M is inflated; uncompressed size is about half that.
For a no-op fetch of a single branch of Chromium over a slow connection, you just spent 60 MiB of your ISP's bandwidth cap on ref advertisements.

(2.16)
Git Protocol v2

$ GIT_TRACE_PACKET=1 git -c protocol.version=2 ls-remote \  
   https://chromium.googlesource.com/chromium/src master 2>&1 | wc 
49   267   3866

More than 1 line in the response due to protocol chatter & the server advertising a small number of ref prefixes. But still this is, like, 4 IPv6 packets.
Mail Templates
VTL has been deprecated since 2.14; removed in 2.16. Sorry, this one actually created work for you.

**tl;dr:**
* Difficult language to learn, supports arbitrary Java code
* HTML templating features lack sufficient XSS protection
VTL has been deprecated since 2.14; removed in 2.16. Sorry, this one actually created work for you.

tl;dr:
* Difficult language to learn, lots of arbitrary Java code
* HTML templating features lack sufficient XSS protection
I'm probably losing your attention so it's back to screenshots.
Who is familiar with this lovely loading screen?
Now project search is fast even when you don't have the project cache fully loaded. You now only have to size your project cache to the number of projects that are actually used.
What's Next?

Disclaimer: these are my viewpoints; I don't claim to have convinced the community/maintainers that any or all of these good ideas.

Now we don't even have screenshots, it's just big words from here on out.
Faster Development

I'm really excited about getting rid of ReviewDb.

We have been holding ourselves back from developing big new features that require substantial DB support because (surprise!) supporting them in both ReviewDb and NoteDb would be too painful.
A lot of my team's time is spent doing user support, triaging bugs, tracking down Prolog errors in logs, etc. This doesn't scale.

We want to figure out which legacy features are causing the most support burden and provide simpler alternatives.

Fun fact: we had an outage yesterday where the internal Android review server was unusable because their prolog rules grew too big for the rules.maxPrologDatabaseSize config option. What does that even mean?
Permission Backends

Not everyone needs the rich multi-dimensional ACL system of Gerrit. 90% of users only need 10% of the features (WAG).

Can we scale this back? Maybe. Who knows. But now we have at least factored out an interface that we can implement in different ways, e.g. Role-Based Access Control.
Account Backends

Probably our #1 complaint from new users: "I can't add my coworker as a reviewer because they haven't logged in yet."

Problem is that Gerrit needs to have a local numeric account ID associated with each user.

What if it didn't? What if it just treated everyone on your LDAP server (in a specific group) as a user?

What if preferences were stored as attributes on your LDAP account so they could be shared across Gerrit servers?
Customers **love** creating new custom labels to handle different CI verification steps. Some of our customers have dozens.

But this isn't great:

* No structured display of build results; bots have to leave comments.
* UI doesn't scale to 10+ labels.

We can do better.
We are on Polymer 1. That's EOL in 2019. 
Polymer 2+3 are each migrations that will take up the PG team's time. 
The new hotness, recommended by the Polymer team, is LitElement. 
Or maybe we should be moving to some other framework. Welcome to the JavaScript treadmill.
Onboarding

Getting people to grok Change-Ids and interactive rebases is tough.
Right now you kind of have to be a "git expert" to use Gerrit effectively.
We want to flesh out in-product help, tutorials, and better concept documentation.
You can tell we're a bunch of developers because by far the best documentation we have is our REST API docs.
Branch Review?

Gerrit is strongly opinionated that each commit is a completed thought.
But even within this framework, sometimes it makes sense to review multiple commits as a single unit.
The only way to do this today by reviewing merge changes, but they can be tough to create and work with (even more so than normal series).
Do we want something more like GitHub pull requests?
Can we achieve that while keeping Gerrit, Gerrit?
Thanks!

Maintainers are coming up so you can ask questions about these topics or anything else.