Moving Fast and Reducing Risk

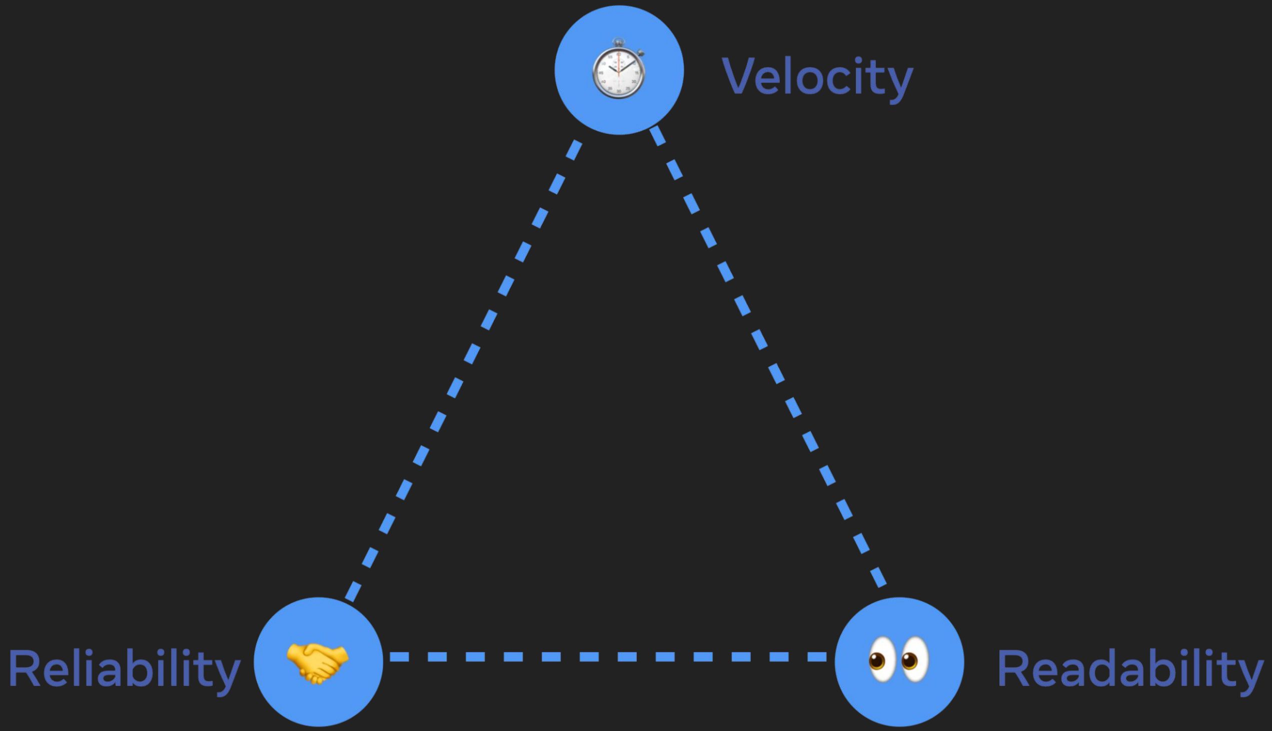
Using LLMs in Release Deployment

DPE 2024 – San Francisco, CA, USA

Rui Abreu

Meta

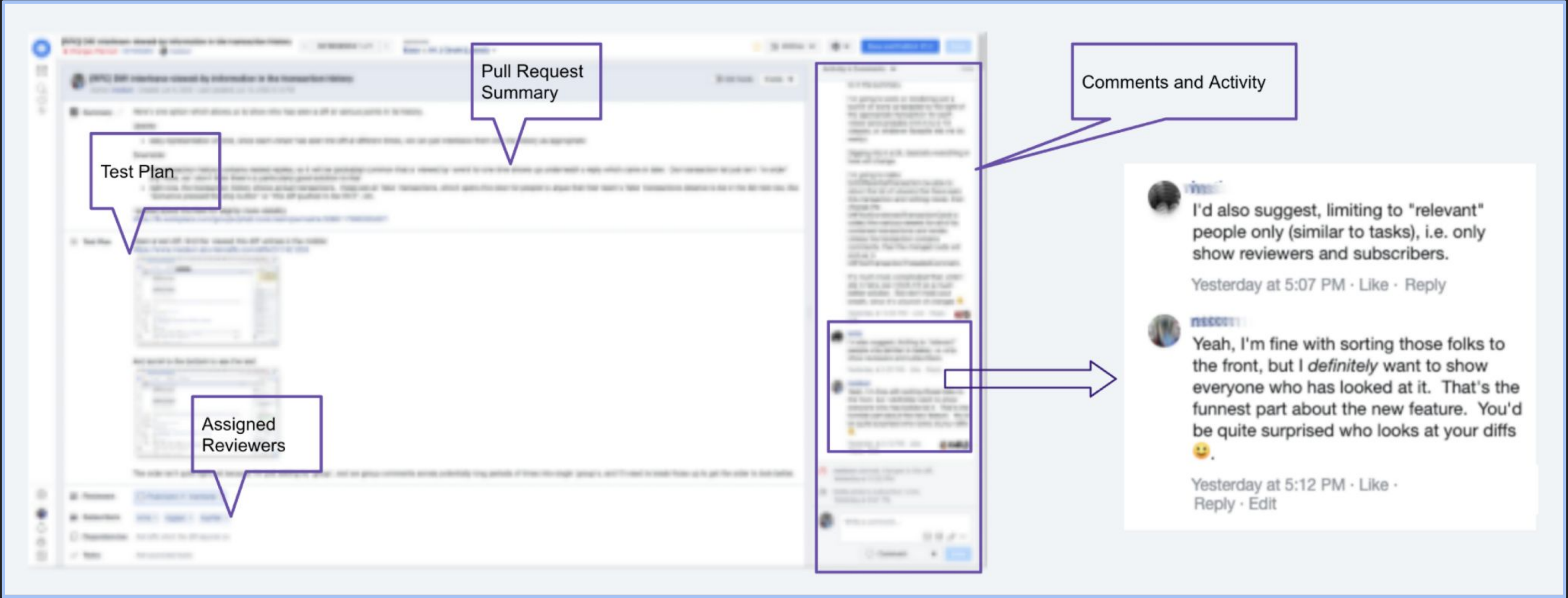
Moving Fast vs. Risk vs. Quality



Release Deployment

{ At Meta scale, **continuous delivery** of new features falls back on the **engineer** writing and reviewing the code. }

Authoring Code @ Meta



[redacted]

I'd also suggest, limiting to "relevant" people only (similar to tasks), i.e. only show reviewers and subscribers.

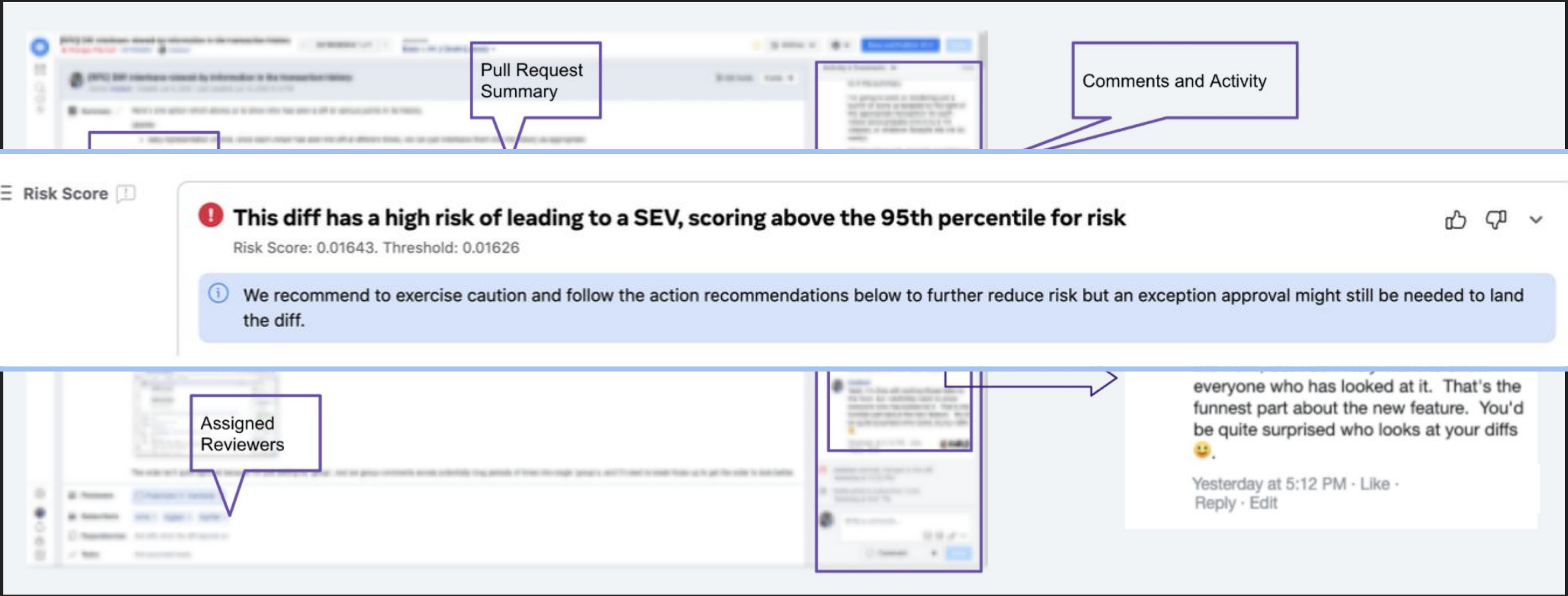
Yesterday at 5:07 PM · Like · Reply

[redacted]

Yeah, I'm fine with sorting those folks to the front, but I *definitely* want to show everyone who has looked at it. That's the funnest part about the new feature. You'd be quite surprised who looks at your diffs 🙂.

Yesterday at 5:12 PM · Like · Reply · Edit

Risk Awareness



Risk Score

This diff has a high risk of leading to a SEV, scoring above the 95th percentile for risk

Risk Score: 0.01643. Threshold: 0.01626

ⓘ We recommend to exercise caution and follow the action recommendations below to further reduce risk but an exception approval might still be needed to land the diff.

everyone who has looked at it. That's the funnest part about the new feature. You'd be quite surprised who looks at your diffs 😃.

Yesterday at 5:12 PM · Like · Reply · Edit

Scenario: Code Freezes

Ensure **stability** and **reliability** during critical periods

Observed during certain periods of the year.

Suspend changes to its production systems to minimize outages (aka SEVs)

🛑Developers can't push new code, and ongoing deployments must be completed before the freeze starts.

⏸️ Unlike traditional code freeze, Meta's code freeze is a code pause or delay where code isn't landed into the monorepo for a short period of time.

The code freeze process has evolved over time, from being based on release engineering team decisions to **individual engineers making the decision to land a diff**.

💥 **Code freezes impacts velocity / productivity!**

How to Deal with Code Freezes

⛔✋ 100% Gating
- 🔴No code is allowed to land!

🚧✋Different gating levels. E.g.,
- 🟢 No gating
- 🟢 Weekend gating (top 5% risky diffs)
- 🟡 Medium impact on end-users (top 10% risky diffs)
- 🔴 High impact on end- users (top 50% risky diffs)

🎯 Increasing developer productivity requires being able to label a code change as being high / low risk!

Risk Prediction Models

- Logistic Regression Models
  - Our baseline regression model captures **18.7%, 27.9%, and 84.6%** of SEVs, while respectively gating the top 5% (weekend), 10% (yellow), and 50% (red) of risky diffs

- BERT-based model
  - StarBERT only captures **0.61×, 0.85×, and 0.81×***

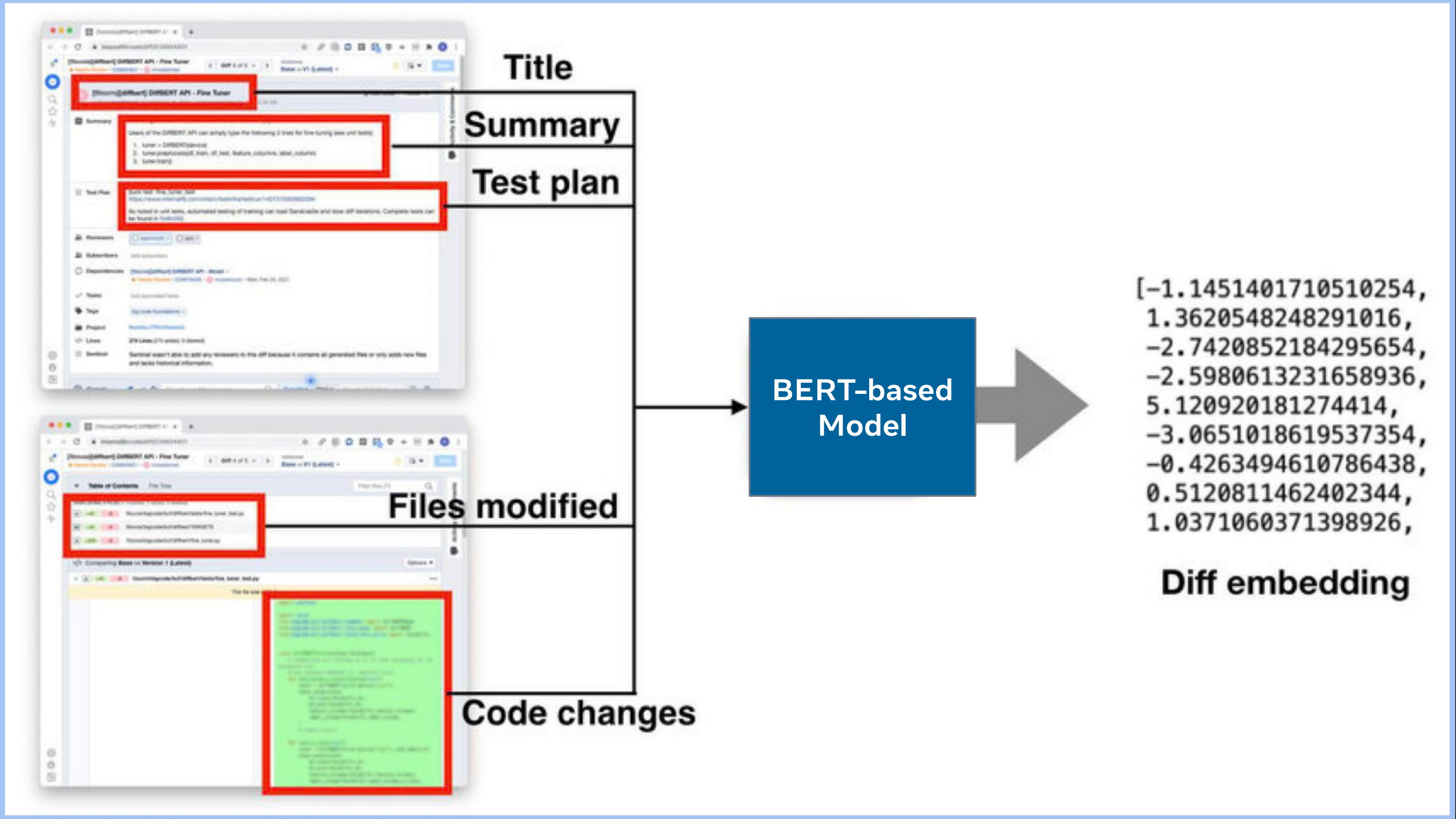
    * as many SEVs as the logistic regression for the 5%, 10%, and 50% gating thresholds, respectively

- Generative LLMs
  - iCodeLlama-34B: **0.58×, 0.65×, and 0.82×**
  - iDiffLlama-13B: **0.65×, 0.81×, and 0.90×**

- Risk-aligned LLMs
  - iCodeLlama-34B: **1.26×, 1.28×, and 0.98×**
  - iDiffLlama- 13B: **1.40×, 1.52×, 1.05×**

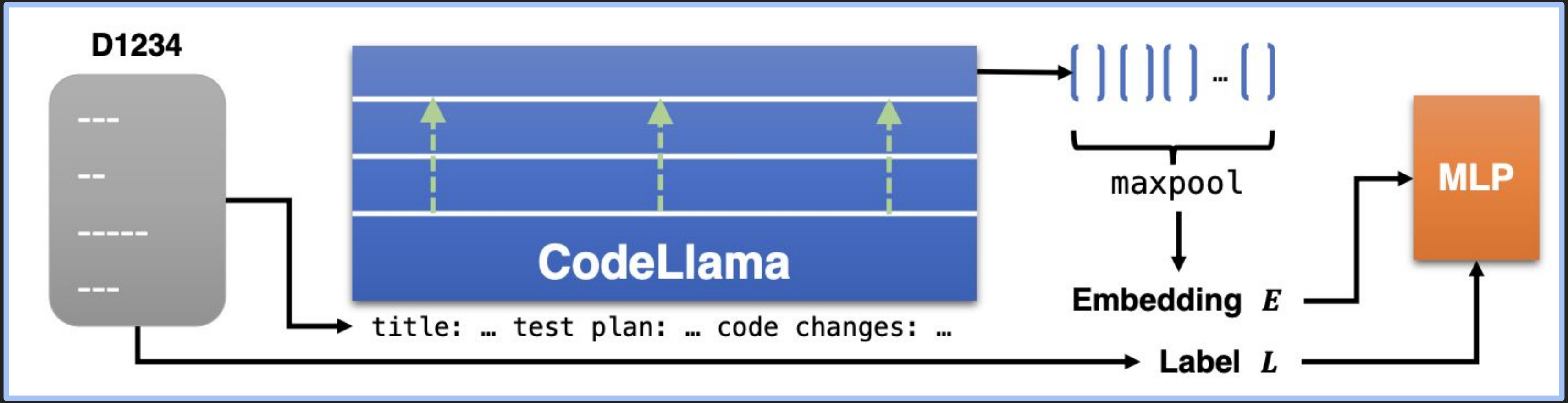
🏆 DiffLama-13B-risk-aligned is the best performing model and is planned to replace the logistic regression model in production.

BERT-based

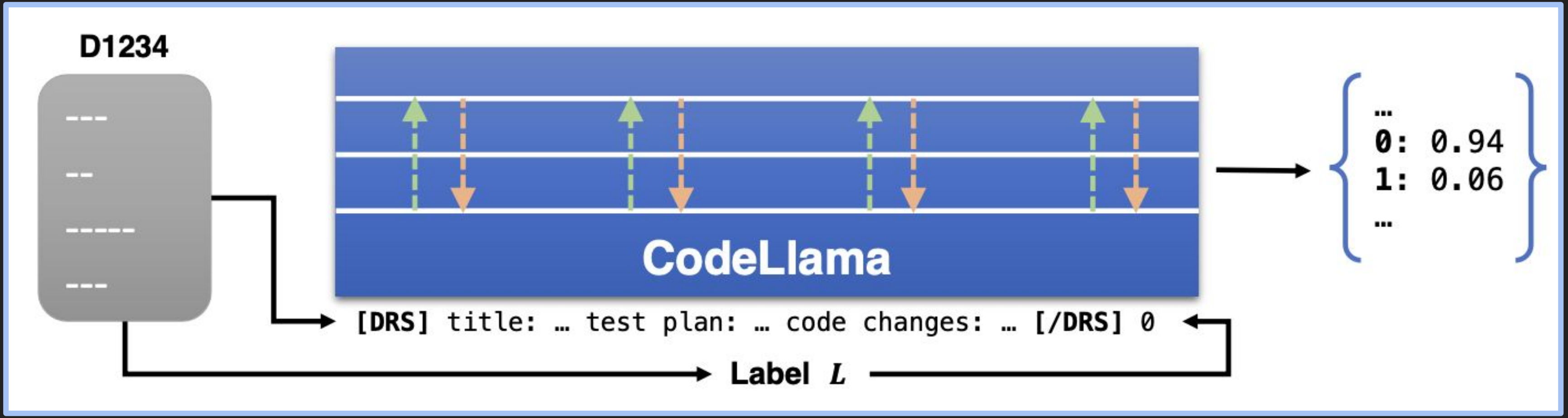


```
[-1.1451401710510254,
 1.3620548248291016,
 -2.7420852184295654,
 -2.5980613231658936,
 5.120920181274414,
 -3.0651018619537354,
 -0.4263494610786438,
 0.5120811462402344,
 1.0371060371398926,
```

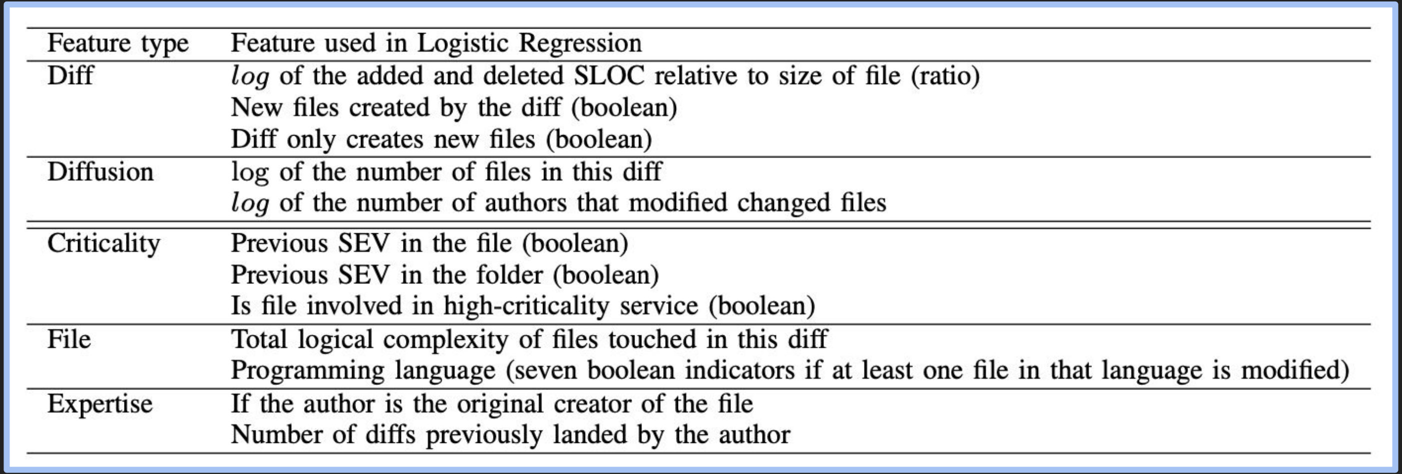
Generative LLMs



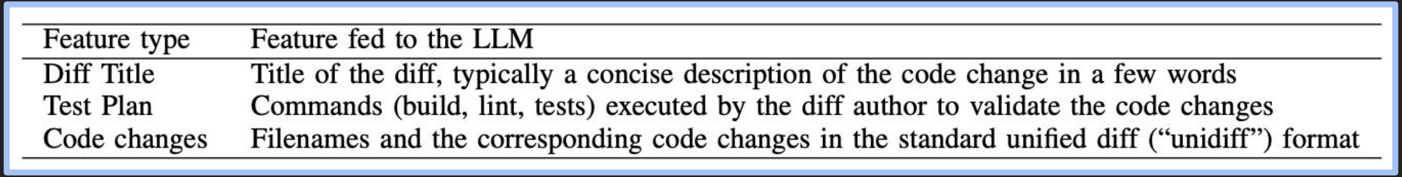
Risk-aligned LLMs



Features for the Models

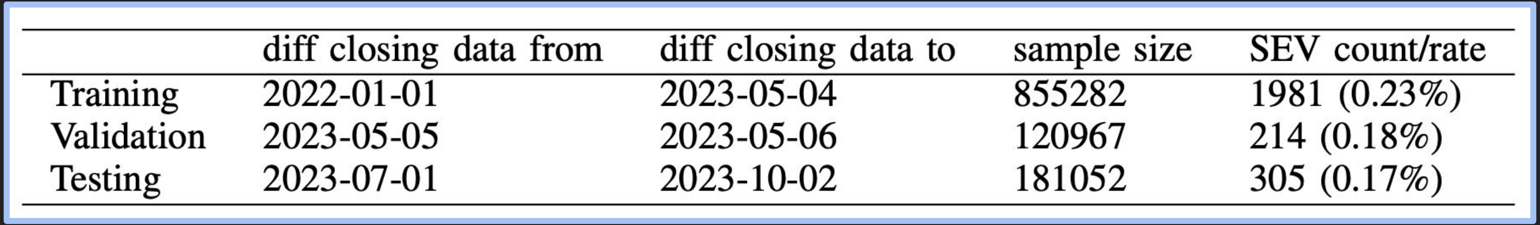


Feature type	Feature used in Logistic Regression
Diff	*log* of the added and deleted SLOC relative to size of file (ratio)
	New files created by the diff (boolean)
	Diff only creates new files (boolean)
Diffusion	log of the number of files in this diff
	log of the number of authors that modified changed files
Criticality	Previous SEV in the file (boolean)
	Previous SEV in the folder (boolean)
	Is file involved in high-criticality service (boolean)
File	Total logical complexity of files touched in this diff
	Programming language (seven boolean indicators if at least one file in that language is modified)
Expertise	If the author is the original creator of the file
	Number of diffs previously landed by the author



Feature type	Feature fed to the LLM
Diff Title	Title of the diff, typically a concise description of the code change in a few words
Test Plan	Commands (build, lint, tests) executed by the diff author to validate the code changes
Code changes	Filenames and the corresponding code changes in the standard unified diff (“unidiff”) format

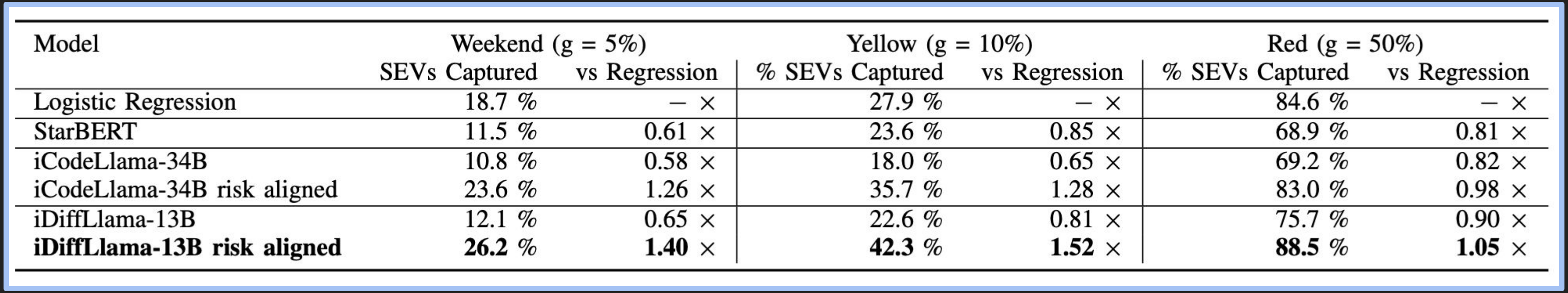
Dataset



	diff closing data from	diff closing data to	sample size	SEV count/rate
Training	2022-01-01	2023-05-04	855282	1981 (0.23%)
Validation	2023-05-05	2023-05-06	120967	214 (0.18%)
Testing	2023-07-01	2023-10-02	181052	305 (0.17%)

⚠️ Extremely imbalanced dataset (rare events!)
🔍 Hence optimizing for recall (and not precision)

Detailed Results



Model	Weekend (g = 5%)		Yellow (g = 10%)		Red (g = 50%)	
	SEVs Captured	vs Regression	% SEVs Captured	vs Regression	% SEVs Captured	vs Regression
Logistic Regression	18.7 %	– ×	27.9 %	– ×	84.6 %	– ×
StarBERT	11.5 %	0.61 ×	23.6 %	0.85 ×	68.9 %	0.81 ×
iCodeLlama-34B	10.8 %	0.58 ×	18.0 %	0.65 ×	69.2 %	0.82 ×
iCodeLlama-34B risk aligned	23.6 %	1.26 ×	35.7 %	1.28 ×	83.0 %	0.98 ×
iDiffLlama-13B	12.1 %	0.65 ×	22.6 %	0.81 ×	75.7 %	0.90 ×
iDiffLlama-13B risk aligned	**26.2** %	**1.40** ×	**42.3** %	**1.52** ×	**88.5** %	**1.05** ×

Conclusions

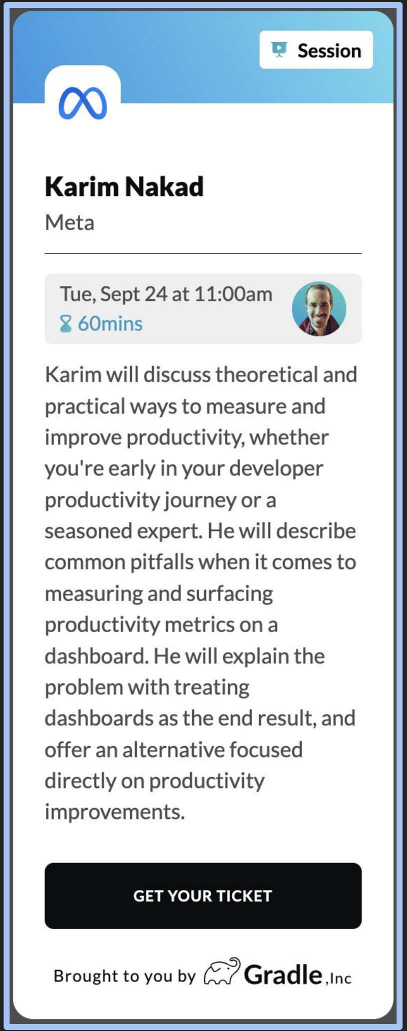
Discussed (ML-based) approaches to code freeze

- 👨‍💻that will improve engineering productivity via unfreeze
- 🚦 by only gating changes that are likely to lead to SEVs

We have shown that the use of ML models can significantly improve the accuracy of diff risk scoring, which can help developers make more informed decisions about which diffs to gate.

🔬Results

- 👌Logistic regression outperformed the RoBERTa-based models.
- 👌The generative LLM models showed promising results
- ✅ iDiffLlama-13B, when risk aligned, model capturing the most SEVs among all models tested.



Session

Karim Nakad

Meta

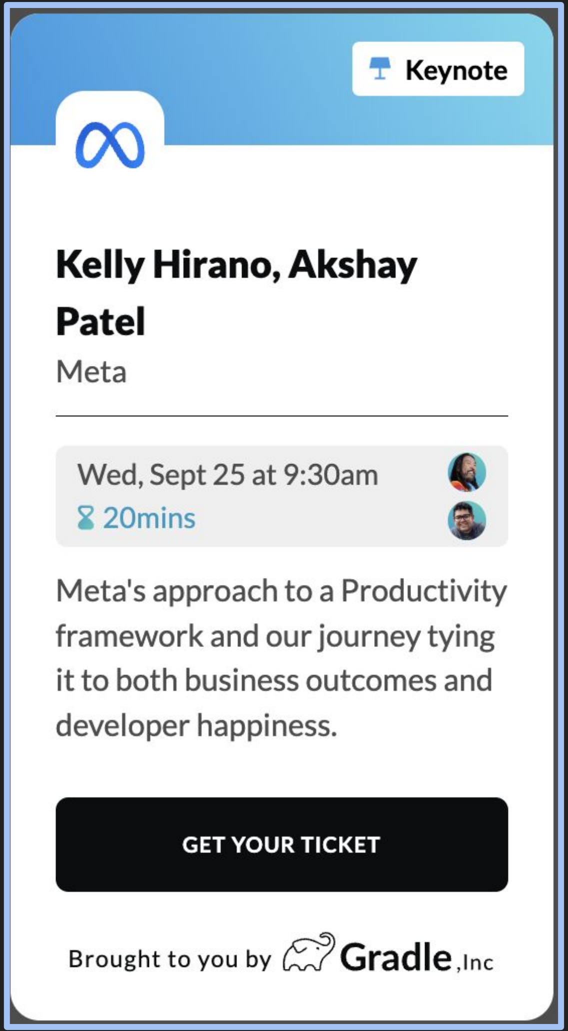
Tue, Sept 24 at 11:00am

60mins

Karim will discuss theoretical and practical ways to measure and improve productivity, whether you're early in your developer productivity journey or a seasoned expert. He will describe common pitfalls when it comes to measuring and surfacing productivity metrics on a dashboard. He will explain the problem with treating dashboards as the end result, and offer an alternative focused directly on productivity improvements.

GET YOUR TICKET

Brought to you by Gradle, Inc



Keynote

Kelly Hirano, Akshay Patel

Meta

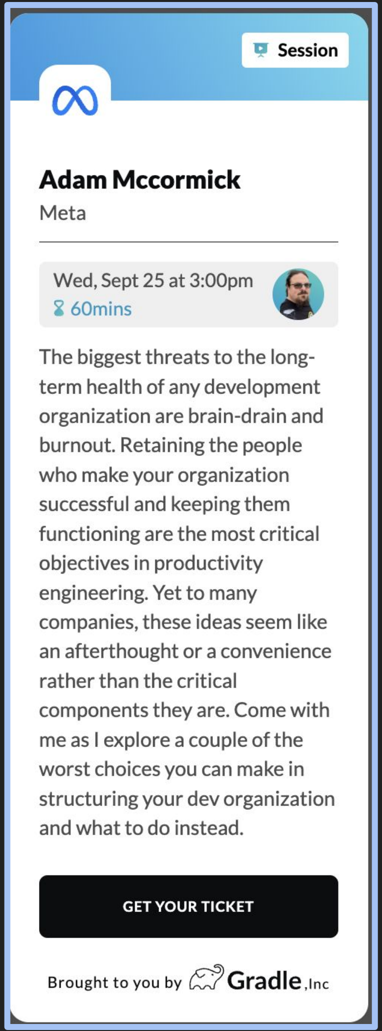
Wed, Sept 25 at 9:30am

20mins

Meta's approach to a Productivity framework and our journey tying it to both business outcomes and developer happiness.

GET YOUR TICKET

Brought to you by Gradle, Inc



Session

Adam Mccormick

Meta

Wed, Sept 25 at 3:00pm

60mins

The biggest threats to the long-term health of any development organization are brain-drain and burnout. Retaining the people who make your organization successful and keeping them functioning are the most critical objectives in productivity engineering. Yet to many companies, these ideas seem like an afterthought or a convenience rather than the critical components they are. Come with me as I explore a couple of the worst choices you can make in structuring your dev organization and what to do instead.

GET YOUR TICKET

Brought to you by Gradle, Inc

Moving Faster and Reducing Risk

Using LLMs in Release Deployment

DPE 2024 – San Francisco, CA, USA

Rui Abreu