

How Uber Built AI Agents That Save 21,000 Developer Hours with LangGraph | LangChain Interrupt



LangChain
119K subscribers

Subscribe

944



Share

Clip

Save



45,467 views Jun 10, 2025

Hear from Sourabh Shirhatti and Matas Rastenis on Uber's Developer Platform Team in their LangChain Interrupt talk. They dive into how they're building agentic tools serving an organization of 5,000 developers working with hundreds of millions of lines of code.



Uber's Scale

\$177 billion

Annualized run-rate gross bookings

33 million

Trips per day

15,000+

Cities

>400 million

Lines of code

100k

Commits per month

5000+

Developers

Public figures for the quarter ended December 31, 2024

AI Dev Products at Uber

Products

AutoCover:
Unit test generator

Validator:
In-IDE Code convention
enforcement

uReview:
Code review in CI

Other Dev Tools

Reusable AI tooling

Shared tools

Shared agents

Frameworks

Uber opinionated framework: LangFx

Shared Infra

Vector DB

AI API Gateway

LangGraph

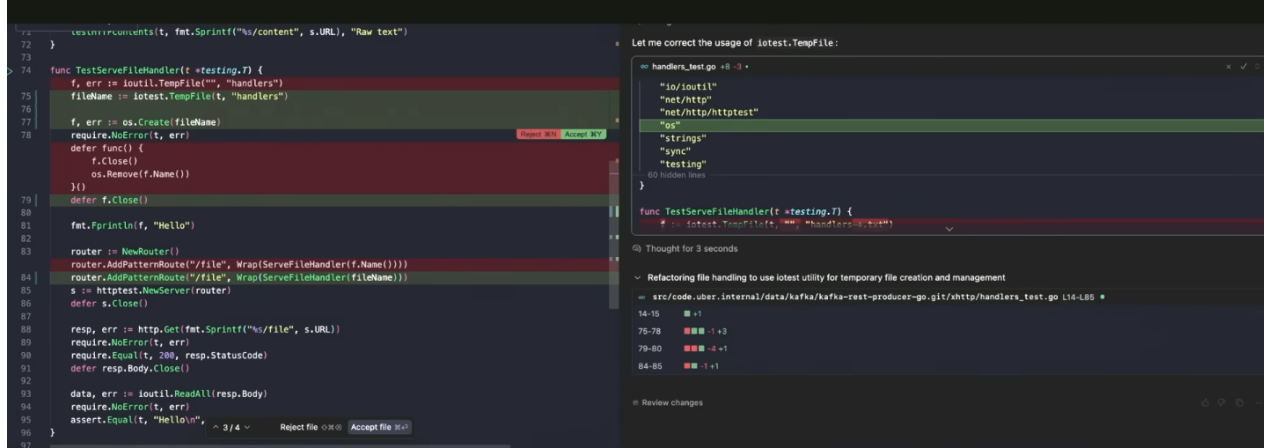
Other shared
infra

Products: Validator

```
38
39
40 Tests should use t.TempFile(), t.TempDir() or "code.uber.internal/base/testing/iotest" for creating temporary files and directories in tests, as these
41 functions automatically clean up the temporary files and directories after the test is complete. Do not use ioutil.TempFile() or os.MkdirTemp() as they
42 do not clean up the temporary files and directories.
43
44 Replace the use of ioutil.TempFile with t.TempFile(), which integrates with Go's test cleanup and ensures temporary files are removed even if the test
45 fails. For example, use:
46
47 f := t.TempFile()
48
49 instead of:
50
51 f, err := ioutil.TempFile("", "handlers")
52 require.NoError(t, err)
53 defer func() {
54     f, err := ioutil.TempFile("", "handlers")
55     require.NoError(t, err)
56     defer func() {
57         f.Close()
58         os.Remove(f.Name())
59     }()
60
61     fmt.Fprintln(f, "Hello")
62
63     router := NewRouter()
64     router.AddPatternRoute("/file", Wrap(ServeFileHandler(f.Name())))
65     s := httptest.NewServer(router)
66     defer s.Close()
67 }
```

This is effectively a LangGraph agent that we built an IDE around, it calls an LLM and returns the insight back as mouseover help to the developer during coding

Products: Validator



Products: AutoCover

```
30 client devpodapi.DevpodAPIAPIClient
31
32
33 func (g >devpodGateway) GetDevpods(ctx context.Context, user string, region string, labels map[string]string)
34 all := (user == "")
35
36 resp, getDevpodsErr := g.client.GetDevpods(ctx, >devpodapi.GetDevpodsRequest{
37     UserID: user,
38     All: all,
39     Region: region,
40     Labels: mapper.FromSpecLabelsToProto(labels),
41 })
42
43 // devpod-api-proxy retrieves devpods from all regions.
44 // If at least one region is down, it returns an error along with devpods from healthy regions.
45 if getDevpodsErr != nil && resp.GetInfos() == nil {
46     return nil, getDevpodsErr
47 }
48
49 result := make([]*entity.Devpod, 0, len(resp.GetInfos()))
50 infos, err := mapper.FromProtoToInfosList(resp.GetInfos())
51 if err != nil {
52     return nil, err
53 }
54
55 for _, info := range infos {
56     devpod := entity.Devpod{
57         Spec: info.Spec,
58         RegionID: g.getRegionID(info.Pod.Region),
59         Pod: info.Pod,
60         Lifecycle: info.Lifecycle,
61     }
62     result = append(result, devpod)
63 }
64
65 return result, getDevpodsErr
66
67
68 func (g >devpodGateway) FindDevpod(ctx context.Context, user string, region string, name string) (*entity.Devpod,
69 name, region, err := g.resolveDevpodName(user, region, name)
70 if err != nil {
71     return entity.Devpod(), errors.Wrap(err, "gateway finding devpod")
72 }
73
74 var devpods []*entity.Devpod
75 devpods, err = g.getDevpods(ctx, user, region, nil)
76 if err != nil {
77     return entity.Devpod(), errors.Wrap(err, "gateway finding devpod")
78 }
79 }
```

INTERRUPT

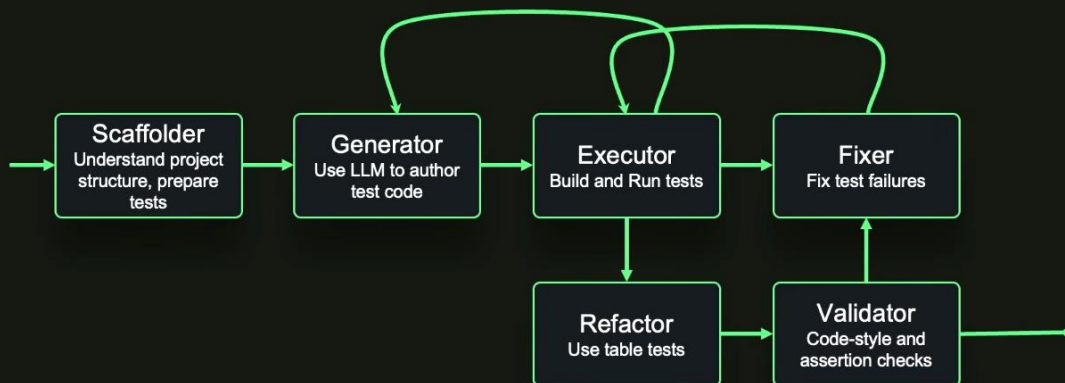
This uses a bunch of domain expert agents in the back that helps to auto-generate test code for the developer.

Products: AutoCover

INTERUPT

```
src > code.uber.internal > devexp > devpod > devpod-cl > gateway > gateway_test.go > ...
1 package gateway
2
3 import {
4     // Import dependencies
5     "context"
6     "errors"
7     "fmt"
8     "testing"
9
10    "code.uber.internal/devexp/devpod/devpod-cli/config"
11    "code.uber.internal/devexp/devpod/devpod-cli/entity"
12    "code.uber.internal/devexp/devpod/devpod-common/devpod"
13    "github.com/stretcher/testify/assert"
14    "github.com/stretcher/testify/require"
15    "go.uber.org/mock/gomock"
16    devpodapi "gooprotos/code.uber.internal/devexp/devpod/devpod-api"
17    devpodapimock "mock/gooprotos/code.uber.internal/devexp/devpod/devpod-api/devpod-api-mock"
18 }
19
20 func TestStubTakeBase[OutputCoverageFile]() {
21     // Generated by AutoCover
22     func TestGetRegionDNS_WhenAutoCover(t *testing.T) {
23         tests := []struct {
24             name string
25             regions []config.DevpodRegion
26             regionName string
27             expectedDNS string
28         }{
29             {
30                 name: "region found in configuration",
31                 regions: []config.DevpodRegion{
32                     {Name: "devpod-pln", DNS: "devpod-pln.dns"},
33                     {Name: "devpod-ald", DNS: "devpod-ald.dns"},
34                 },
35                 regionName: "devpod-pln",
36                 expectedDNS: "devpod-pln.dns",
37             },
38             {
39                 name: "region not found in configuration",
40                 regions: []config.DevpodRegion{
41                     {Name: "devpod-pln", DNS: "devpod-pln.dns"},
42                     {Name: "devpod-ald", DNS: "devpod-ald.dns"},
43                 },
44                 regionName: "devpod-unknown-region",
45                 expectedDNS: "devpod-unknown",
46             },
47             {
48                 name: "empty region name",
49                 regions: []config.DevpodRegion{
50                     {Name: "devpod-pln", DNS: "devpod-pln.dns"},
51                 },
52             },
53         }
```

AutoCover



Product gallery

Picasso Genie AI Beta

Beta release, possible inaccuracies.

Welcome to Picasso Genie! I'm here to streamline your workflow by answering questions and helping you draft support requests. To get started, please make a selection or ask a question.

- 🔗 Ask about the platform.
- 📄 Submit a support request.
- 📖 Review Picasso documentation.
- 🔍 Find an automation.

Type your message...

Uber Assistant Builder Beta

Chat with Assistants built with Uber connectivity or create your own. [🔗](#)

Chat Store

Search...

+ Add filter

score-bot

accesshealth

Helps create awareness about your current security behavior, reflecting your individual cybersecurity...

+9066 users

whober-assistant

uber-assistant

ML Platform

A generalist assistant that is good at answering Uber specific questions. It has realtime read access to Jira...

+936 users

sop-assistant-grocery-curatio...

Product gallery

Picasso Genie AI Beta

Beta release, possible inaccuracies.

Welcome to Picasso Genie! I'm here to streamline your workflow by answering questions and helping you draft support requests. To get started, please make a selection or ask a question.

- 🔗 Ask about the platform.
- 📄 Submit a support request.
- 📖 Review Picasso documentation.
- 🔍 Find an automation.

Type your message...

732

```
if serviceDeployment.State == entity.ServiceDeploymentStateDeployed || serviceDeployment.State == entity.ServiceDeploymentStateConnected {
```

rohit.rathi

Done

Nit: We can iterate in reverse order to get results in fewer iterations.

psg-genai-bot

Done

uReview suggests the following change to address the comment above:

```
+++
@@ -730,5 +730,6 @@ func checkServiceRedeployment(requestedService string,
    deployedServices []entity
    if deployedService.Name == requestedService {
-       for _, serviceDeployment := range deployedService.ServiceDeploymentInfo {
-           if serviceDeployment.State == entity.ServiceDeploymentStateDeployed ||
serviceDeployment.State == entity.ServiceDeploymentStateConnected {
+           for i := len(deployedService.ServiceDeploymentInfo) - 1; i >= 0; i-- {
+               if deployedService.ServiceDeploymentInfo[i].State ==
entity.ServiceDeploymentStateDeployed || deployedService.ServiceDeploymentInfo[i].State
== entity.ServiceDeploymentStateConnected {
+                   serviceAlreadyDeployed = true
+                   break
+               }
+           }
+       }
    }
```

Provide fix-correctness feedback: <Correct>, <Partially Correct>, <Incorrect>.

Run this command to apply fix: (Contact #ureview-support for support)

ureview D14343879:391375255

Learnings - Design Principles

🧠 Specialized = Smarter

Purpose-Built Agents Outperform General Foundation Models

- Better context retention
- Less ambiguity
- Fewer hallucinations

🧩 Simple, Reliable Nodes

Bounded Responsibilities + Deterministic Nodes Improve Reliability

- Deterministic tools (e.g., linters) ensure consistency
- Clear boundaries simplify development and debugging

⚡ Fast & Scalable Dev

Reusable Agent Nodes Accelerate Development

- Solve scoped problems and reuse across graphs
- Faster composition of new flows
- Reuse nodes; iterate faster

Learnings - Strategy Benefits

👉 Encapsulation Boosts Collaboration

- Agent nodes can be developed and tested in isolation
- Teams work in parallel on isolated agent nodes
- Horizontal scale-out allows you to tackle more problems

🔄 Graphs Improve Human + AI Workflows

- The graph naturally represents complex workflows like human reasoning
- Graph structure reveals real-world bottlenecks and dependencies
- Process improvements for AI also streamline human operations
- E.g., faster infra helps both automated and manual execution