Code Reviewer Plugin

Justin Horton
Workday
justin.horton@workday.com

Tamás Szabó
itemis
tamas.szabo@itemis.de
This presentation may contain forward-looking statements for which there are risks, uncertainties, and assumptions. If the risks materialize or assumptions prove incorrect, Workday’s business results and directions could differ materially from results implied by the forward-looking statements. Forward-looking statements include any statements regarding strategies or plans for future operations; any statements concerning new features, enhancements or upgrades to our existing applications or plans for future applications; and any statements of belief. Further information on risks that could affect Workday’s results is included in our filings with the Securities and Exchange Commission which are available on the Workday investor relations webpage: www.workday.com/company/investor_relations.php

Workday assumes no obligation for and does not intend to update any forward-looking statements. Any unreleased services, features, functionality or enhancements referenced in any Workday document, roadmap, blog, our website, press release or public statement that are not currently available are subject to change at Workday’s discretion and may not be delivered as planned or at all.

Customers who purchase Workday, Inc. services should make their purchase decisions upon services, features, and functions that are currently available.
Workday & MPS

- Applications built for 10+ years in XpressO (XO)
  - High-level, form-based metadata
  - Clicking & copy/paste

- YP built in MPS as a replacement
  - Goals: Improve application performance & developer productivity
  - Familiar functional programming concepts
  - Built-in constructs for common application logic
• Multiple teams implementing DSLs
  – 3 core language DSLs
  – 2 teams starting on their (integrated) DSLs

• Beginning language adoption

• Rapidly growing MPS dev team
  – ~4 devs a year ago up to ~15 now
Team growth meant more time spent on pull request feedback.

Source diff vs. projectional diff challenging for targeted comments:
- Stylistic feedback noise.
Improving the Code Review Experience
Initial Attempts

**MPS Node URLs**
- Reduce developer time to find and address feedback

**Lexer**

**Reducer**

**Linter**
- Reduce pull request noise by automating style fixes
Improving the Code Review Experience

Existing Tools

**mbeddr Code Review**
- Inline commenting
- Carries review information along with the model

**Screenshot Bitbucket Plugin**
- Doesn’t modify the model
- Screenshots make inline commenting impossible

**Bitbucket Plugin**
- Inline commenting

Feedback stored with the pull request

???
• Post feedback within MPS
  – Optionally associate with particular nodes
  – Distributed via the pull request
• View feedback inline
  – Achieved with mbeddr conditional editors
• Mirror other pull request info
• Works with any MPS DSL
Code Reviewer Plugin
Code Reviewer Plugin Architecture
Flux

http://facebook.github.io/flux/docs/in-depth-overview.html#content
Flux

```
Store (com.workday.mps.flux.core)
  ActivityStore (com.workday.mps.review.store)
  ReviewStore (com.workday.mps.review.store)
  SettingsStore (com.workday.mps.review.store)
  UiProxyStore (com.workday.mps.review.store)
```

http://facebook.github.io/flux/docs/in-depth-overview.html#content
Flux

http://facebook.github.io/flux/docs/in-depth-overview.html#content
Evaluation

• Open source
  – Find us at https://github.com/Workday/mps-code-reviewer

• Use for MPS dev teams within Workday

• Extend to language consumers within Workday

• Incorporate feedback and iterate
Future Work

- Support GitHub

- Support other pull request actions
  - Approve, decline, needs work
  - Like comments
  - Reply to comments

- Improve commenting/refresh UX

- Make it prettier
  - Replace Swing
  - Render markup and images
Conclusion

**Workday & MPS**

- Applications built for 10+ years in XpressO (XO)
  - Form-based metadata
  - High level
- YP built in MPS as a replacement
  - Goals: Improve application performance & developer productivity
  - Familiar functional programming concepts
  - Built-in constructs for common application logic

**Improving the Code Review Experience**

**Existing Tools**

- mbeddr Code Review
  - Inline commenting
  - Carries review information along with the model
- Bitbucket Plugin
  - Screenshots make inline commenting impossible
  - Doesn’t modify the model
- Feedback stored with the pull request

**Problem**

**Code Review Workflow**

- Team growth meant more time spent on pull request feedback
- Source diff vs. projectional diff challenging for targeted comments
  - Stylistic feedback noise

**Code Reviewer Plugin Architecture**

https://github.com/Workday/mps-code-reviewer