

Lightweight Architecture Decision Records

Aurelijus Banelis



VilniusPHP 0x44
2018-07-12



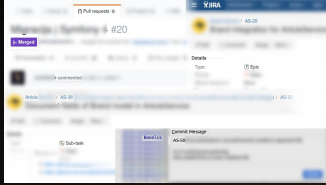
Aurelijus Banelis

Software developer

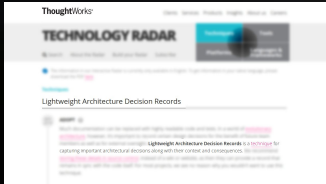
aurelijus.banelis.lt
aurelijus@banelis.lt

PGP 0x320205E7539B6203
130D C446 1F1A 2E50 D6E3
3DA8 3202 05E7 539B 6203

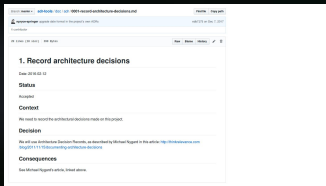




Problem



Proposal



Summary

Problem

12/09/16
12/09/16
12/09/16
12/09/16

Reikės dokerio

History: README.md

2
7
5
3
3
7
7
3

◆ Update README.md


◆ fixed misstype

12/09/16
12/09/16
12/09/16
12/09/16

Reikės dokerio

History: README.md

- Update README.md
- fixed misstype

 Pull requests 0

#20

Merged

commented

Article / AS-38

/ AS-50


Details

Type:  Epic

Commit Message

Banelis

AS-50

 Sub-task

Pull requests 0

AS-20

#20

Merged

Details

Type:

Epic

commented

Article / AS-38

/ AS-50

Commit Message

Banelis

AS-50

Sub-task



Proposal (solution)

TECHNOLOGY RADAR

Home About the Radar Build your Radar Subscribe



The information on our Technology Radar is currently only available in English. To get information in your native language, click on the language icon.

Adopting

Lightweight Architecture Decision Records



ADOPTING

Much documentation can be replaced with highly readable code and tests. In a world of **continuous delivery**, however, it's important to record certain design decisions for the benefit of future team members as well as for external consulting.

Lightweight Architecture Decision Records is a **technique** for capturing important architectural decisions along with their context and consequences.

Lightweight Architecture Decision Records instead of a set of articles, so that they can provide a record that remains in sync with the code base. For most projects, we don't believe why you shouldn't want to use this technique.

 npryce-springer upgrade date format in the project's own ADRs

edb7175 on Dec 7, 2017

1 contributor

20 lines (10 sloc) | 399 Bytes

Raw

Blame

History



1. Record architecture decisions

Date: 2016-02-12

Status

Accepted

Context

We need to record the architectural decisions made on this project.

Decision

We will use Architecture Decision Records, as described by Michael Nygard in this article: <http://thinkrelevance.com/blog/2011/11/15/documenting-architecture-decisions>

Consequences

See Michael Nygard's article, linked above.

1. Meet regularly

Date: 2018-07-12

Status

Accepted

Context

Problem: It is hard to find speakers for every VilniusPHP meetup. Speakers do not want to present to empty room.

Alternatives:

- Give JetBrains licenses to Speakers – no big impact seen, when not giving
- Give branded T-Shirts to speakers – works for old members, but hard for newcomers
- Buy beer for attendees – seems to be working, but half of attendees do not go to afterparty
- Advertise via other meetups – there a few going from Kaunas, but hard to scale

Decision

Keeping meetups at a regular basis is the cheapest (price+effort) way to keep Vilnius PHP community active.

Consequences

Book venue in advance. If there are no speakers – meet just to drink some beer.

Summary

Way to document architecture decisions

- **Low effort to try**
- **Value in the future**

Lightweight Architecture Questions? Decision Records

Aurelijus Banelis



Lightweight Architecture Decision Records

Thank you

Aurelijus Banelis



VilniusPHP 0x44

2018-07-12