



Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

FAKULTÄT
FÜR MATHEMATIK, INFORMATIK
UND NATURWISSENSCHAFTEN



ADR, MADR, eADR

Architekturentscheidungen sichtbar und nachvollziehbar gestalten

10.03.2026

Dr. Oliver Kopp

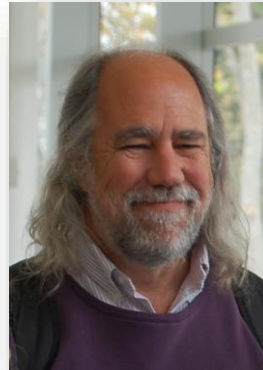
Über Oliver Kopp

- Studium der Softwaretechnik
- Promotion im Bereich Web Services (WS-*) / Business Process Management
- 5 Jahre Industrie (Mercedes-Benz AG, Nokia Bell Labs)
- Derzeit Vertretung der Professur für [Softwareentwicklungs- und konstruktionsmethoden](#)
- Java seit Version 1.2
- [JabRef](#) Maintainer seit 15+ Jahren

Software-Architektur: Definition

“Architecture represents the **significant design decisions** that shape a **system**, where significant is measured by **cost of change.**”

[Grady Booch \(2006\) “On design”](#) cited in: Frank Buschmann, Kevlin Henney, Douglas C. Schmidt (2007): *Pattern-Oriented Software Architecture*. p. 214.



Architektur und getroffene Entscheidungen



Y-Statements

In the context of <use case/
user story/component ctx>,

facing <non-functional concern c>,

we decided for <option o1>

and neglected <options o2 to on>



to achieve <quality q>,
accepting downside <consequence c>.

AD aspect (attribute)	Y-Statements (ABB, [24])	IEEE 42010 (Template V2.2)	IBM UMF AD Table	Tyree/Akerman	Bredemeyer Key Decisions	Nygard ADRs	arc42 Hruschka/Starke
<i>ID</i>	(Id)	Unique Identifier	ID	(in D-Header)	/	(part of name)	(Section #)
<i>Outcome</i>	we decided for	Statement of the decision	Decision (Made)	Decision	Approach	Decision	Decision
<i>Requirements trace (FRs, NFRs)</i>	/	Correspondence or linkage to concerns	(Derived requirements)	Related requirements	Business drivers, technical drivers	/	/
<i>Accountability (Role, Person)</i>	/	Owner of the decision	/	/	/	/	/
<i>Software architecture viewpoint trace</i>	In the context of	Correspondence or linkage to elements	/	Related artifacts	/	/	/
<i>Why-answers</i>	(optional "because" half sentence)	Rationale (linked entity)	Justification	Argument	Conclusion	/	(Question under Decision)
<i>Decision drivers</i>	facing	Forces, constraints	/	(Constraints)	Benefits, Drawbacks	Context	Constraints
<i>Assumptions</i>	/	Assumptions	Assumptions	Assumptions	/	/	Assumptions
<i>Options</i>	and neglected	Considered Alternatives	Alternatives	Positions	/	/	Considered Alternatives
<i>Problem</i>	/	/	Issue or Problem	Issue	/	/	Problem
<i>Decision dependencies</i>	/	(not in template, but in standard)	Related decisions	Related decisions	/	/	/
<i>Categorization, classification</i>	/	/	Subject Area, Topic	Group(ing)	/	/	(Decision Topic)
<i>Name</i>	/	/	Name	(in D-Header)	<<key decision>>	Title	(Section heading)
<i>State of AD making</i>	/	not in template, but in standard	(not in published example)	Status	/	Status	/
<i>Impact</i>	to achieve, accepting that	not in template, but in standard	Implications	Implications	Issues/ Considerations	Consequences	/
<i>Other entries</i>	/	Timestamps, Citations	Motivation	Notes, Related principles	Notes, Drivers realized	/	/
Element count	6	9 (template), 11 (standard)	13	14	9 (plus 1-2 in header)	5	5 (with 14 questions)
Scoping help (which ADs to capture?)	/	Yes	(not in table template, not published)	(anecdotal In article)	2001 white paper	/	(ASRs mentioned)
Size (page or word limit) or other hints	1 (long) sentence per Y-statement	/	not published	(example is half a page, table form)	/	1-2 pages per ADR	to be ordered by importance
Publication year	2012	2011	1998 (internal)	2005	2005	2011	2012

Source:
[Olaf Zimmermann et al.: Architectural Decision Guidance across Projects, WICSA 2015](#)

Entscheidungstreiber

Nahe am Quelltext

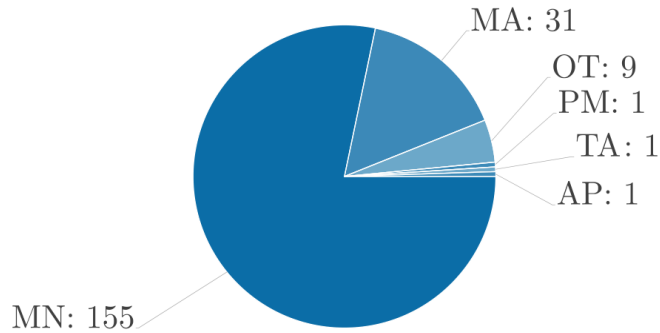
Durch strukturierte Erfassung hohe Qualität

Ergebnis

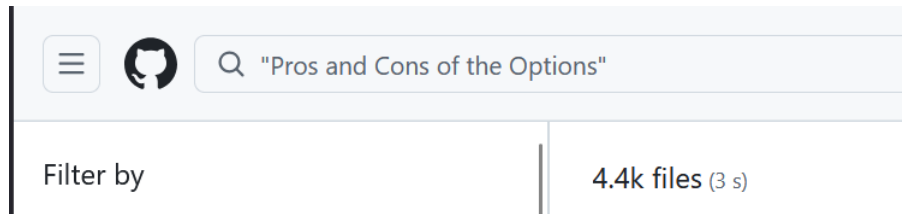
Markdown Architectural Decision Records

[Erstes Release](#) am 9. September 2017

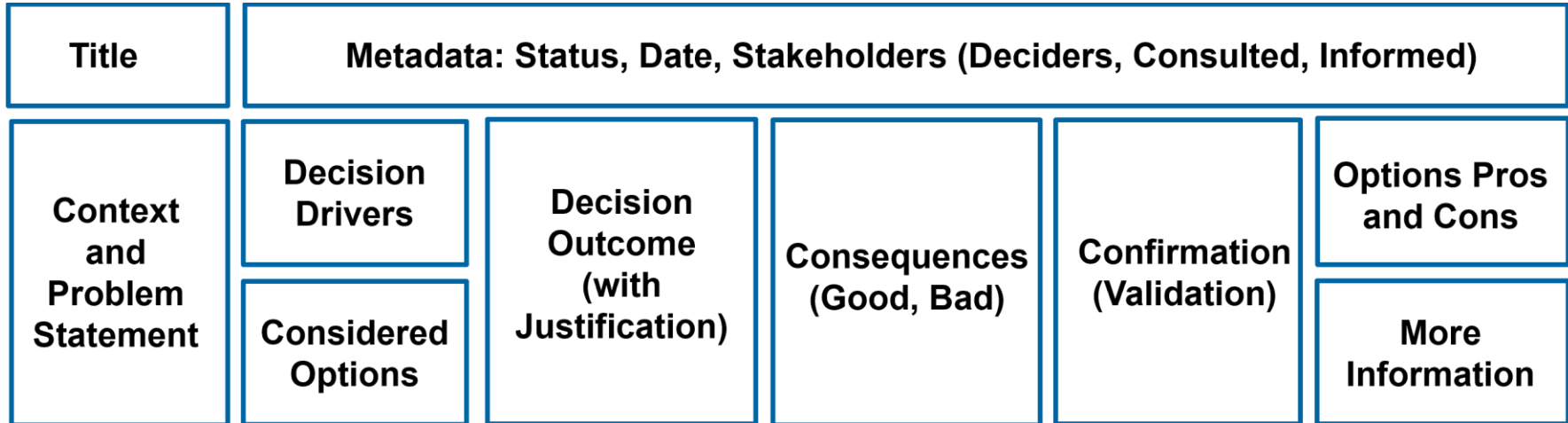
Market Share



- 16% Marktanteil auf GitHub
- Bei Repositorys mit mehr als 10 ADRs
- Stand: 2023
- Quelle:
[G. Buchgeher, S. Schöberl, V. Geist, B. Dorninger, P. Haindl, and R. Weinreich, "Using Architecture Decision Records in Open Source Projects—An MSR Study on GitHub," *IEEE Access*, vol. 11, pp. 63725–63740, 2023, doi: 10.1109/access.2023.3287654.](#)



MADR-Bestandteile



MADR Bare Template

```
Preview Code Blame 44 lines (25 loc) · 522 Bytes
1 ---
2 status:
3 date:
4 decision-makers:
5 consulted:
6 informed:
7 ---
8
9 √ # <!-- short title, representative of solved problem and found solution -->
10
11 √ ## Context and Problem Statement
12
13
14
15 √ ## Decision Drivers
16
17 * <!-- decision driver -->
18
19 √ ## Considered Options
20
21 * <!-- option -->
22
```

```
23 √ ## Decision Outcome
24
25 Chosen option: "", because
26
27 √ ### Consequences
28
29 * Good, because
30 * Bad, because
31
32 √ ### Confirmation
33
34
35
36 √ ## Pros and Cons of the Options
37
38 √ ### <!-- title of option -->
39
40 * Good, because
41 * Neutral, because
42 * Bad, because
43
44 ## More Information
```

MADR als *.md direkt unter docs/decisions

The screenshot shows an IDE with a project named 'JabRef'. The file explorer on the left shows a 'docs' directory containing a 'decisions' subdirectory. The main editor displays the file '0009-use-plain-junit5-for-testing.md'. The content of the file is as follows:

```
13
14 * Plain JUnit5
15 * Hamcrest
16 * AssertJ
17
18 ## Decision Outcome
19
20 Chosen option: "Plain JUnit5", because comes out best
21 \ (see below).
22
23 ### Positive Consequences
24
25 * Tests are more readable
26 * More easy to write tests
27 * More readable assertions
28
29 ### Negative Consequences
30
31 * More complicated testing leads to more complicated
32 assertions
33
34 ## Pros and Cons of the Options
35
36 ### Plain JUnit5
37
38 Homepage: <https://junit5
```

On the right side of the IDE, there is a 'Gradle' panel showing the project structure:

```
Gradle
├── JabRef (root)
│   ├── Tasks
│   ├── Run Configurations
│   └── jabgui
│       ├── Tasks
│       │   └── application
│       │       ├── run
│       │       ├── runMacos-15
│       │       ├── runMacos-15-intel
│       │       ├── runShadow
│       │       ├── runUbuntu-22.04
│       │       ├── runUbuntu-22.04-arm
│       │       └── runWindows-latest
│       ├── build
│       ├── dependency-analysis
│       ├── distribution
│       ├── documentation
│       ├── help
│       ├── java modules
│       ├── other
│       └── reporting
```

Tooling: ADR-Manager

The screenshot shows the ADR-Manager web application. The top navigation bar is blue and contains the logo 'ADR-Manager', the text 'Editor Mode: basic professional', and a 'DISCONNECT' button. On the left, there is a sidebar with a list of ADRs, each with a trash icon. The main content area shows an ADR titled 'Plain JUnit5 for advanced test assertions'. Below the title, there is a note: 'Changing the title changes the file name. Do not use special characters.' There are fields for 'Last Update', 'Status: No status', and 'Deciders'. A 'Technical Story' field is present. Below this is a section titled 'Context and Problem Statement' with a text area containing the text: 'How to write readable test assertions?' and 'How to write readable test assertions for advanced tests?'.

Considered Options ?

This section lists 'Plain JUnit5' as a considered option. It includes a 'Description' field with the following content:

Homepage: <https://joel-costigliola.github.io/assertj/>

Example:

```
```java
assertThat(markdownFormatter.format(source))
 .contains("Markup
")
 .contains("list item one")
 .contains("list item 2")
 .contains("> rest")
 .doesNotContain("\n");
...
```
```

Below the description, there is a 'Good, because ... ?' section with three bullet points:

- offers fluent assertions
- allows partial string testing to focus on important parts
- assertions are more readable

Embedded ADRs – io.github.adr:e-adr:2.0.0

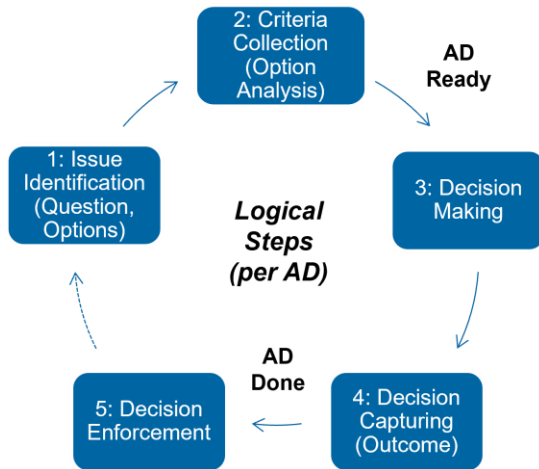
- @ADR(9)



```
@MADR(value = 1,
    title = "Implement entire business logic layer as POJO",
    contextAndProblem = "Need to select a single consistent ComponentImplementationParadigm' for entire
alternatives = {"POJO", "JEE EJB", "Spring Bean"},
chosenAlternative = "POJO",
justification = "Because it is a company-wide policy, see http://www.iaas.uni-stuttgart.de/institut/relatedDecisions = {2, 3})
public class MADRAnnotationDemoClass {
    public int sampleBusinessLogic(String input) {
        return 42;
    }
}
```

Themenfelder

Vorgehensweise



<https://ozimmer.ch/practices/2023/12/01/ADDefinitionOfReady.html>



Review von ADRs

| | |
|-------------------------------------|---|
| Roles | <i>Peer, Coach/Mentor</i>
<i>Direct Stakeholders External Design Authority</i> |
| Review Advice (Do) | <i>Deliver what is asked for, prioritize, document scope and goals of the review</i>

<i>Refer to quality attributes, acknowledge context, be concrete and factual, thorough, focused</i>

<i>Problem-solution style, impressions but no interpretations, good and bad, fair and polite</i>

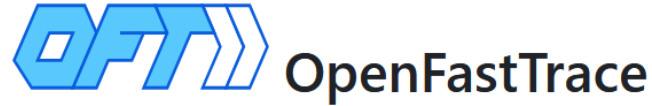
<i>Make comments resolvable, offer help with resolution, review the review</i> |
| Reviewer Pledge | <i>Manage scope, content, style (professional, constructive)</i>
<i>Avoid anti-patterns, use checklist, ensure actionability</i>
<i>Review like you want to be reviewed</i> |
| Related Advice (Other Posts) | <i>ASR Test, MADR or Y-Statements, Definition of Done (AD)</i>
<i>ADR creation practices and anti-patterns</i> |

| |
|---|
| Anti Patterns (Don't) |
| <i>Pass Through</i> |
| <i>Copy Edit</i> |
| <i>Siding, Dead End</i> |
| <i>Self Promotion, Conflict of Interest</i> |
| <i>Power Game</i> |
| <i>Groundhog Day</i> |
| <i>Offended Reaction</i> |

<https://ozimmer.ch/practices/2023/04/05/ADRReview.html>

Forschung und Entwicklung I

- Anstatt Java-Annotationen:

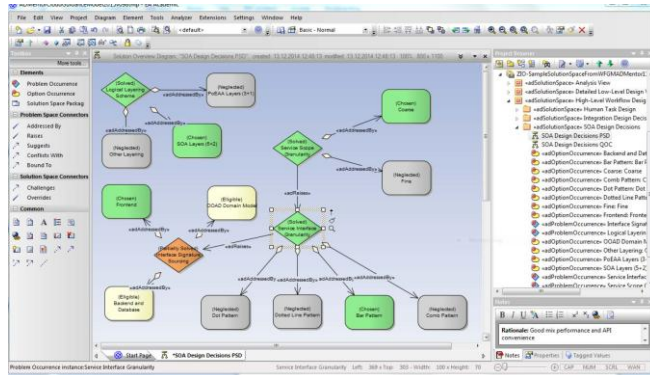


```
// [impl->adr~testing.use-junit~1]
```

- Verbessertes Tooling
 - Web-GUI (ADR Manager)
 - [VS.Code plugin](#)
 - TUI
 - Linting

Forschung und Entwicklung II

- Generalized Architectural Decision Records
 - <https://github.com/adr/gadr-java>
- Guidance Tooling



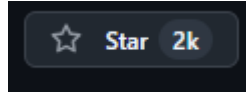
Considered Options

1. Apache Commons CLI
2. Picocli
3. JCommander
4. Airline 2
5. ritopt
6. crest

- Können LLMs „saubere“ ADRs schreiben?

Markdown Architectural Decision Records (“MADR”)

- Optionen strukturiert erfassen
- <https://github.com/adr/madr>



- Danke an [Olaf Zimmermann](#), [Tobias Unger](#), [Justus Bogner](#), [Mathias Schubanz](#), [Sebastian Baltes](#)



Dr. Oliver Kopp

Leveraging Scientific Innovations for Industrial
Application | SDV & BPM Specialist | Open-Source C...

