INNOQ EVENT / 15. FEBRUAR 2018 / DÜSSELDORF

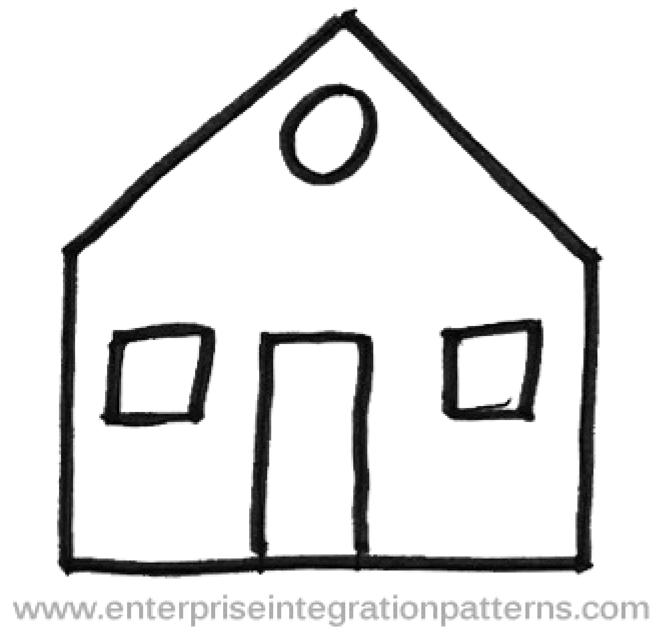
Brief Introduction to Architectural Decision Records

Tammo van Lessen

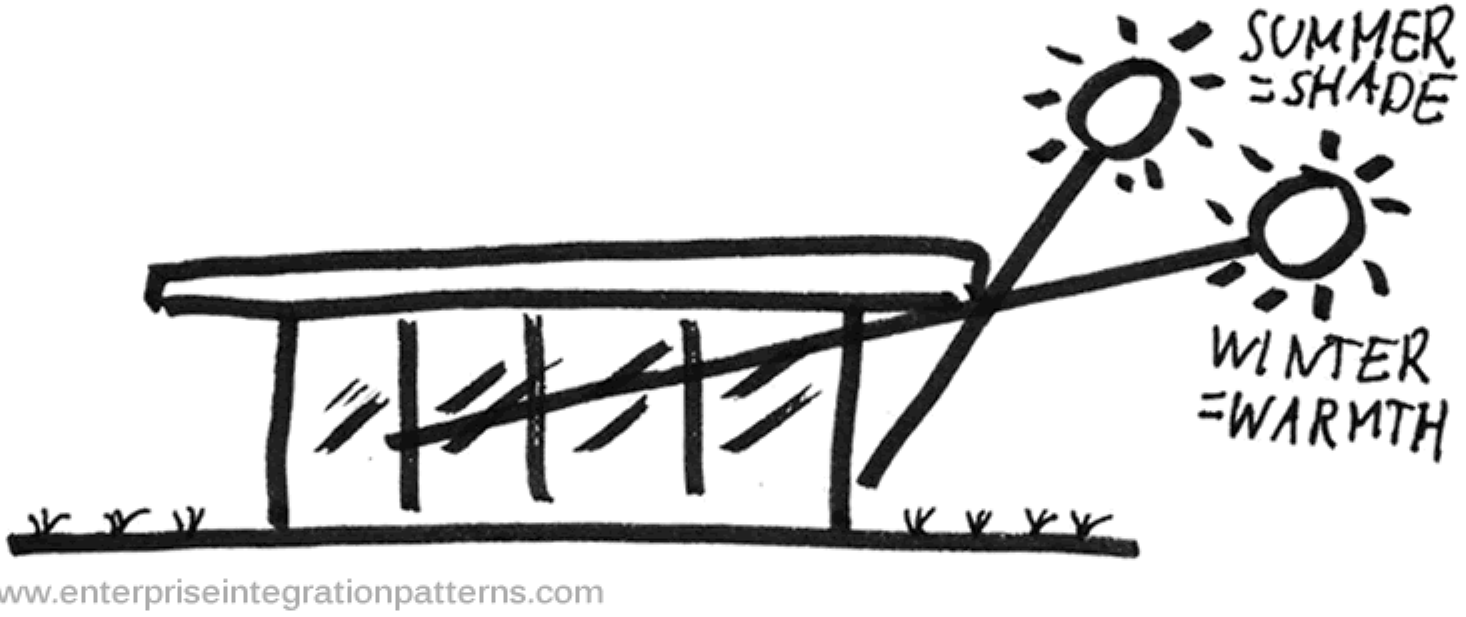
Agenda

- Motivation
- Definition
- Templates
- Tooling

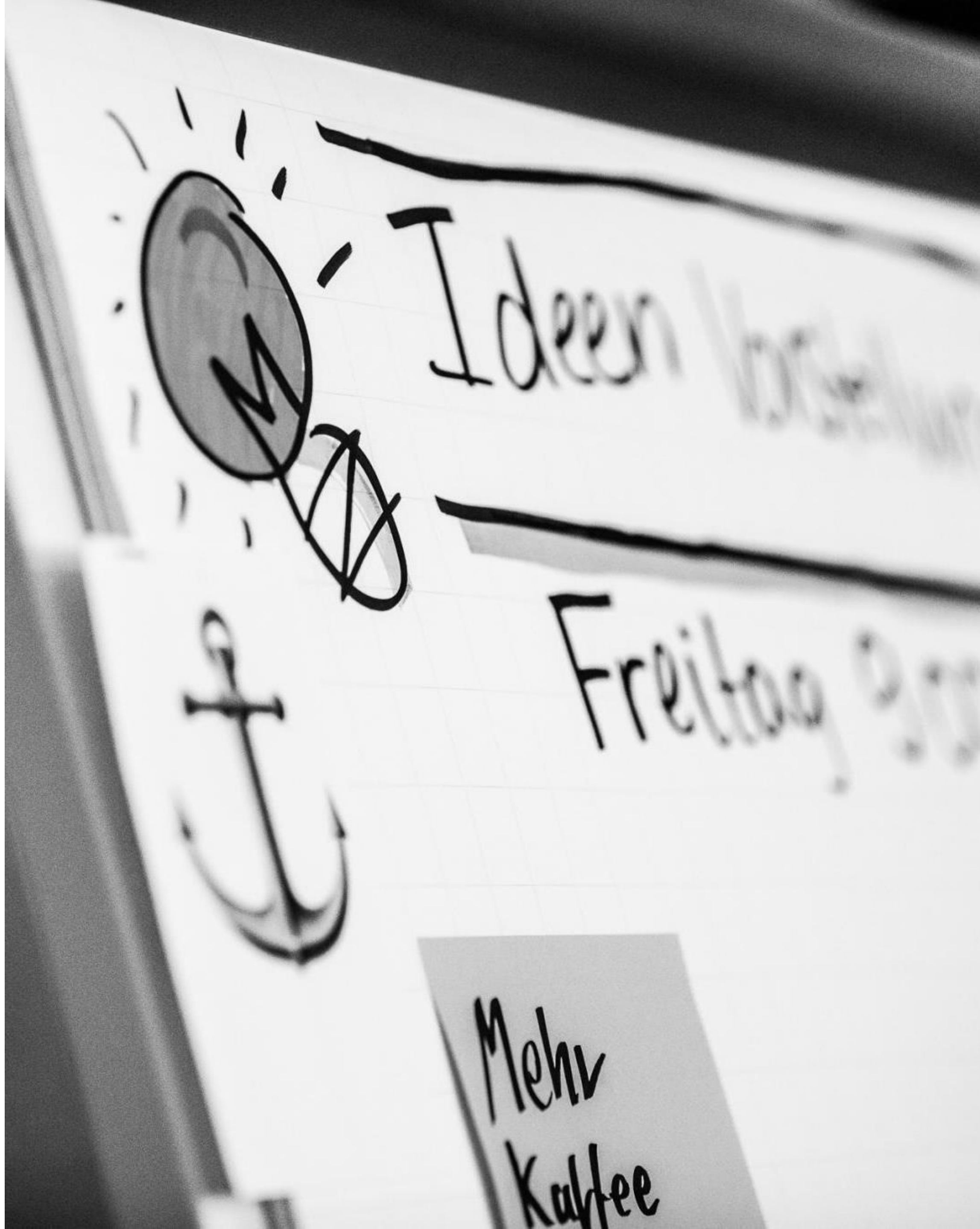
What is Architecture?



What is Architecture?



Decisions & Context



“The major problem with intellectual capital is that it has legs and walks home every day.”

– M. Lindvall, I. Rus

M. Lindvall, I. Rus, R. Jammalamadaka, and R. Thakker. Software Tools for Knowledge Management: A DACS State-of-the-Art Report. Technical report, Fraunhofer Center for Experimental Software Engineering Maryland and The University of Maryland, 2001.

Challenges

- Documentation is an unloved child
- Onboarding / Knowledge Transfer
- Decision drivers are lost frequently
- When to document a decision?
- How to share decision knowledge?
- Are decisions "reuseable"?
    - Context matters…

Problem Space: Architectural Knowledge Management

An **Architecturally Significant Requirement (ASR)** is a requirement that has a measurable effect on a software system's architecture and quality.

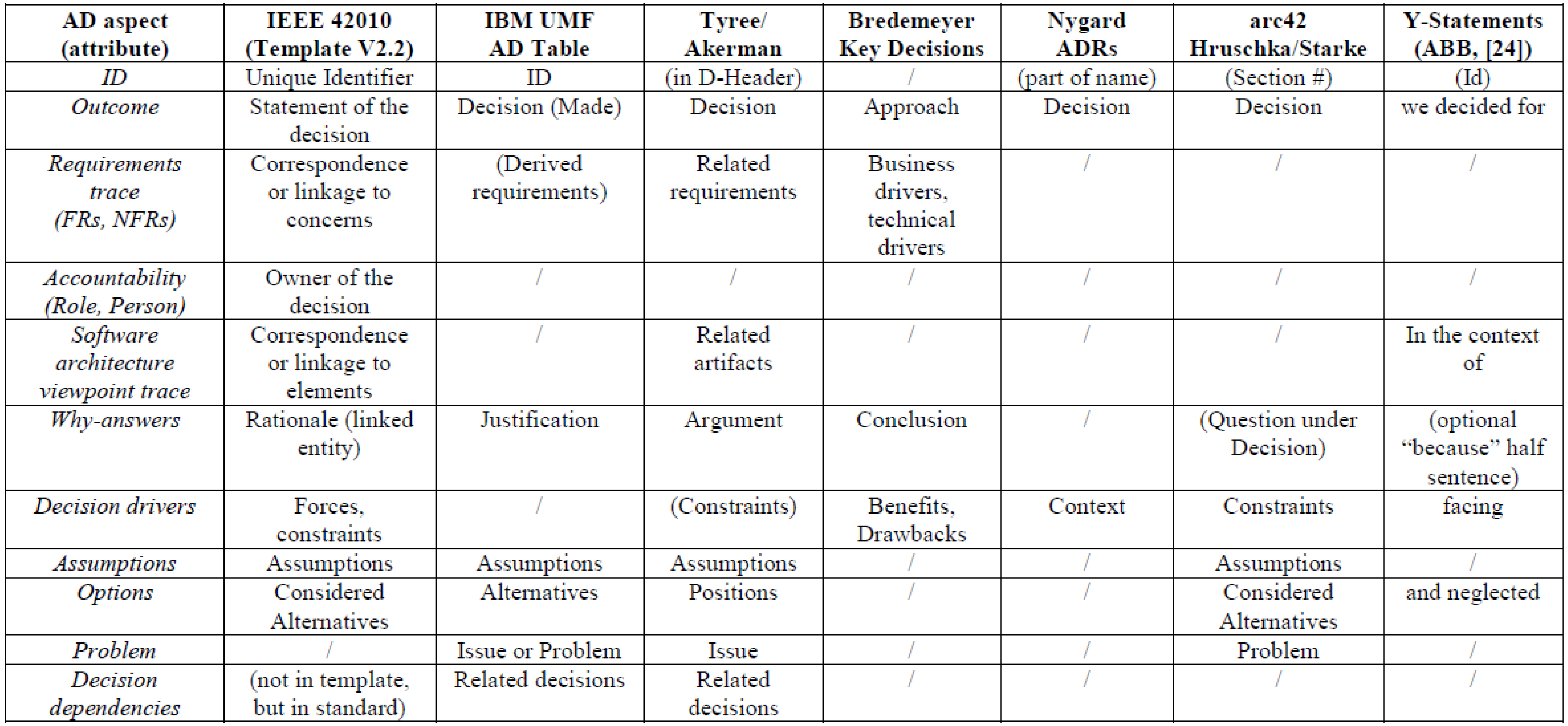
An **Architectural Decision (AD)** is a software design choice that addresses a functional or non-functional requirement that is architecturally significant.

An **Architectural Decision Record (ADR)** captures a single AD, such as often done when writing personal notes or meeting minutes; the collection of ADRs created and maintained in a project constitute its decision log.

All these are within the topic of **Architectural Knowledge Management (AKM)**.

https://adr.github.io / *https://github.com/joelparkerhenderson/architecture_decision_record/*

Problem Space: Architectural Knowledge Management (2)



AD aspect (attribute)	**IEEE 42010 (Template V2.2)**	**IBM UMF AD Table**	**Tyree/ Akerman**	**Bredemeyer Key Decisions**	**Nygard ADRs**	**arc42 Hruschka/Starke**	**Y-Statements (ABB, [24])**
ID	Unique Identifier	ID	(in D-Header)	/	(part of name)	(Section #)	(Id)
Outcome	Statement of the decision	Decision (Made)	Decision	Approach	Decision	Decision	we decided for
Requirements trace (FRs, NFRs)	Correspondence or linkage to concerns	(Derived requirements)	Related requirements	Business drivers, technical drivers	/	/	/
Accountability (Role, Person)	Owner of the decision	/	/	/	/	/	/
Software architecture viewpoint trace	Correspondence or linkage to elements	/	Related artifacts	/	/	/	In the context of
Why-answers	Rationale (linked entity)	Justification	Argument	Conclusion	/	(Question under Decision)	(optional "because" half sentence)
Decision drivers	Forces, constraints	/	(Constraints)	Benefits, Drawbacks	Context	Constraints	facing
Assumptions	Assumptions	Assumptions	Assumptions	/	/	Assumptions	/
Options	Considered Alternatives	Alternatives	Positions	/	/	Considered Alternatives	and neglected
Problem	/	Issue or Problem	Issue	/	/	Problem	/
Decision dependencies	(not in template, but in standard)	Related decisions	Related decisions	/	/	/	/

Zimmermann et al.: Architectural Decision Guidance Across Projects - Problem Space Modeling, Decision Backlog Management and Cloud Computing Knowledge. WICSA 2015: 85-94.

Y-Statements

In the context of <*use case uc*>
and/or <*component co*>,

... facing <*non-functional concern c*>,

We chose <*option* o_1>

and neglected <*option* o_2 *to* o_n>

...to achieve <*quality q*>
... accepting downside <*consequences* $c_1..c_n$>

U. Zdun, R. Capilla, H. Tran, O. Zimmermann, *Sustainable Architectural Design Decisions, IEEE Software, Volume 30, Number 6* (2013). https://www.infoq.com/articles/sustainable-architectural-design-decisions

What are ADRs?

What is the least we can document and still remain effective?

„An architecture decision record is a **short text file** in a format similar to an Alexandrian pattern that describes **a set of forces** and **a single decision** in response to those forces. “ [Nygard2011]

- ADRs to be put as close as possible to the code.
- Build up decision history

What are ADRs? (2)

Alters externally visible system properties

Modifies a public interfaces

Directly influences a high priority quality attribute

Includes or removes a dependency

Direct result of new information about a constraint

Accepts strategic technical debt

Changes the general structures of the system

Forces developers to change their development approach

Extracted from Keeling, M; Runde, J: Architecture Decision Records in Action. https://resources.sei.cmu.edu/asset_files/Presentation/2017_017_001_497746.pdf

ADR Templates

Nygard Template

- Number / Title
- Status
- Context
- Decision
- Consequences

ADR template by Michael Nygard (simple and popular)

ADR template by Jeff Tyree and Art Akerman (more sophisticated)

ADR template for Alexandrian pattern (simple with context specifics)

ADR template for business case (more MBA-oriented, with costs, SWOT, and more opinions)

ADR template MADR

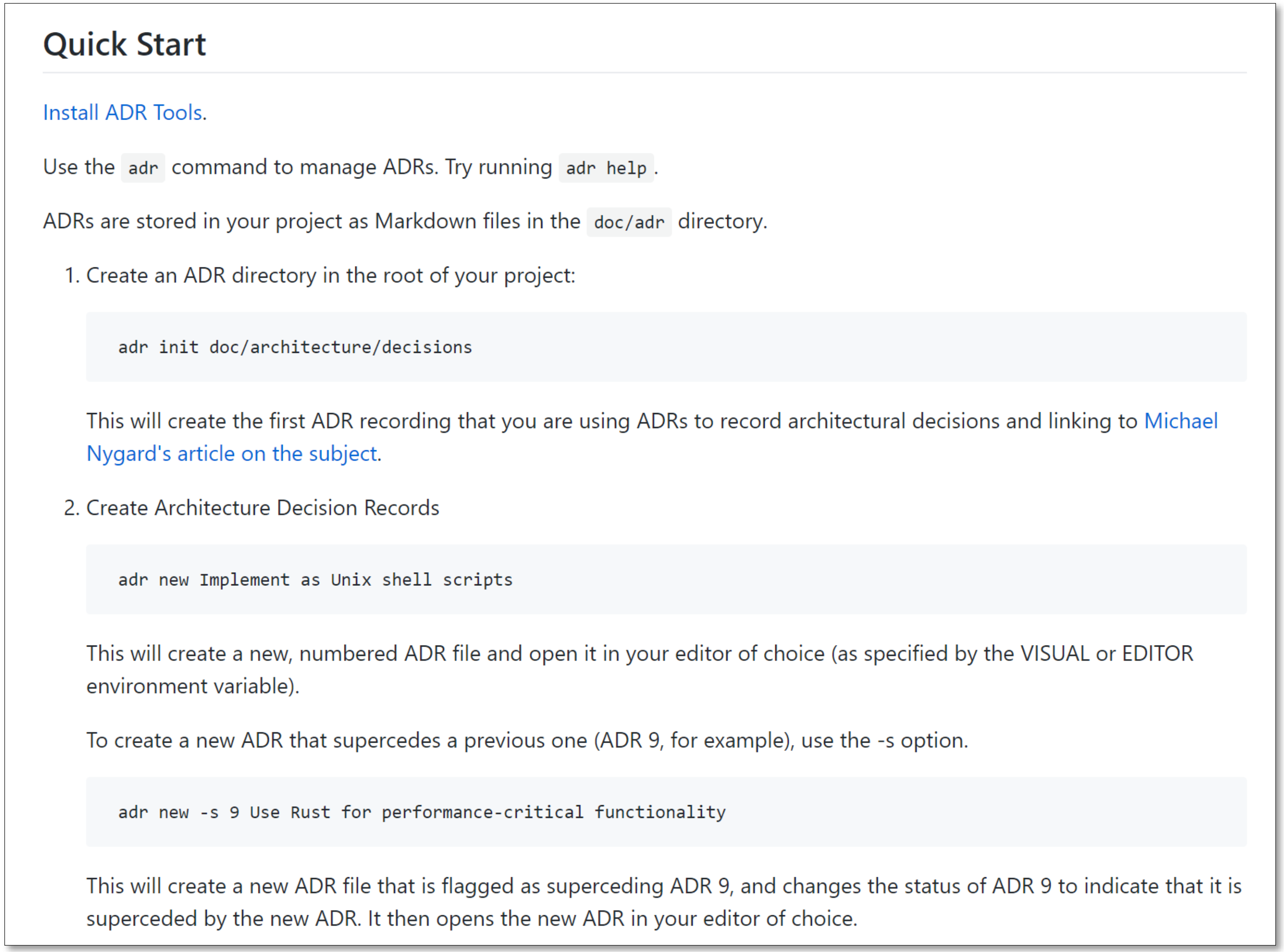
ADR template using Planguage (more quality assurance oriented)

https://github.com/joelparkerhenderson/architecture_decision_record#adr-example-templates / http://thinkrelevance.com/blog/2011/11/15/documenting-architecture-decisions

[ˈmærɚ]?

```
# [short title of solved problem and solution]

User Story: [ticket/issue-number] <!-- optional -->

[context and problem statement]
[decision drivers | forces | facing] <!-- optional -->

## Considered Options
* [option 1]
* [option 2]
* [option 3]
* … <!-- numbers of options can vary -->

## Decision Outcome
Chosen option: [option 1], because [justification. e.g., only
option, which meets k.o. criterion decision driver | which
resolves force force | … | comes out best (see below)].

Positive Consequences: <!-- optional -->
- [e.g., improvement of quality attribute satisfaction,
follow-up decisions required, …]
- ...
Negative consequences: <!-- optional -->
- [e.g., compromising quality attribute, follow-up decisions
required, …]
- …

## Pros and Cons of the Options <!-- optional -->
### [option 1]
* Good, because [argument a]
* Good, because [argument b]
* Bad, because [argument c]
* … <!-- numbers of pros and cons can vary -->

### [option 2]
…
```

adr-tools



Quick Start

Install ADR Tools.

Use the `adr` command to manage ADRs. Try running `adr help`.

ADRs are stored in your project as Markdown files in the `doc/adr` directory.

1. Create an ADR directory in the root of your project:

   ```
   adr init doc/architecture/decisions
   ```

   This will create the first ADR recording that you are using ADRs to record architectural decisions and linking to Michael Nygard's article on the subject.

2. Create Architecture Decision Records

   ```
   adr new Implement as Unix shell scripts
   ```

   This will create a new, numbered ADR file and open it in your editor of choice (as specified by the VISUAL or EDITOR environment variable).

   To create a new ADR that supercedes a previous one (ADR 9, for example), use the -s option.

   ```
   adr new -s 9 Use Rust for performance-critical functionality
   ```

   This will create a new ADR file that is flagged as superceding ADR 9, and changes the status of ADR 9 to indicate that it is superceded by the new ADR. It then opens the new ADR in your editor of choice.

https://github.com/npryce/adr-tools

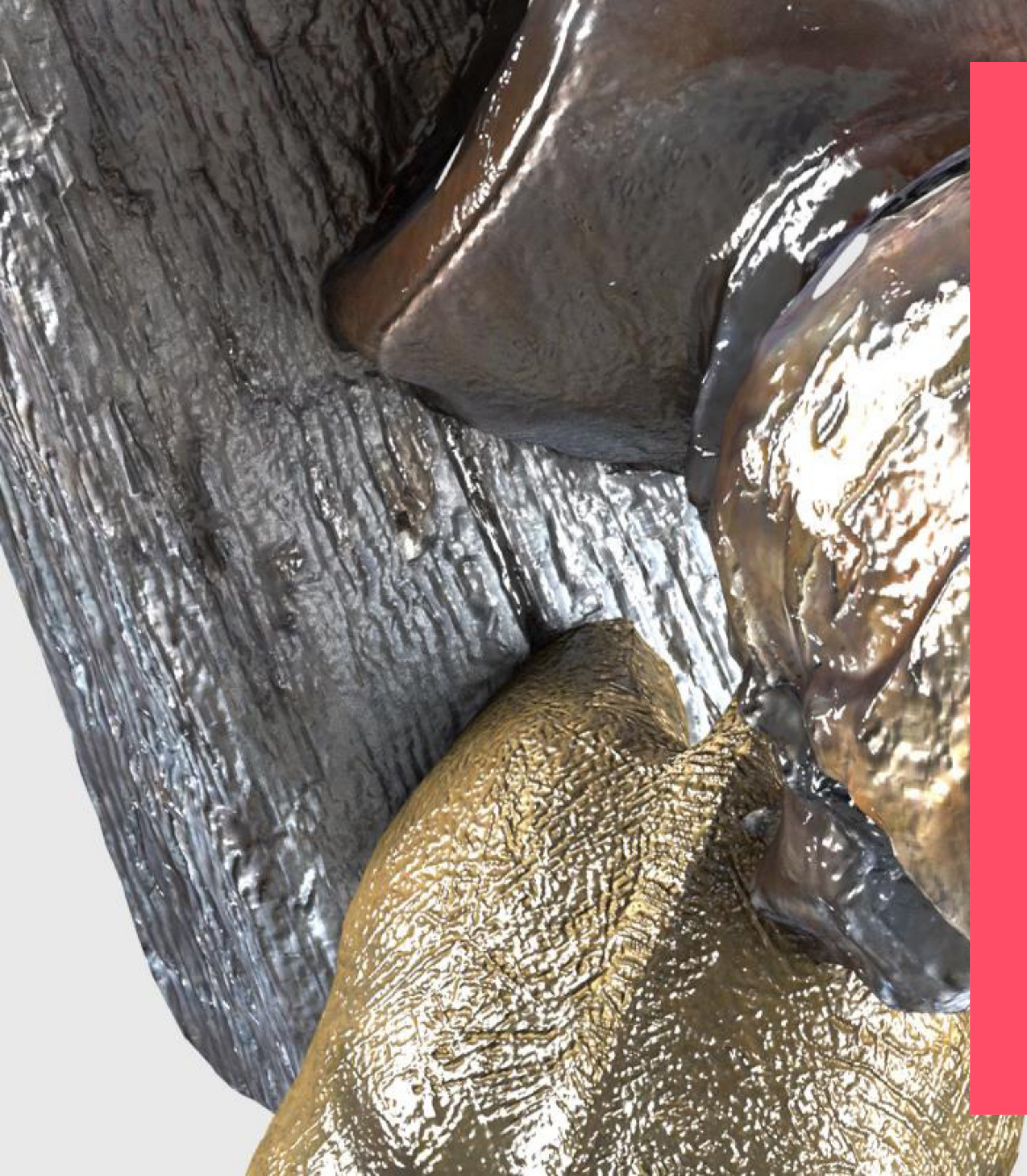
Conclusion & Outlook

ADRs provide a lean and lightweight approach to document architectural decisions.

Lightweight tooling exists to publish and link ADRs easily.

„Decisions required“ vs „Decisions made“

Best practices to deal with Macro / Micro Architecture distinction?



ARCHITECTURAL DECISION RECORDS
FEBRUARY EVENT 2018

Thanks!

Questions?

Tammo van Lessen
tammo.van-lessen@innoq.com

INNOQ