Nick Rozanski
Andy Longshaw
Eoin Woods

Sold!

How to Describe, Explain and Justify your Architecture

Objectives of Today

If you are an architect who has to produce an Architectural Description, then this session will help you answer the following questions:

- What level of detail should I include in my AD?
- Where is the dividing line between the AD and requirements? the AD and other design documents?
- How do I recognise a concern as being architecturally significant and therefore worthy of discussion in my AD?
- What techniques can I use to get my messages across succinctly and strongly?
- What sorts of documentation should I produce?
- How long should my AD be? How do I stop it becoming too wordy and cluttered but still provide the required level of detail?
- How much audit trail should I include (decisions, rationale, alternatives rejected)?
- What documentation tools should I use?
- What works?

This session is a **Workshop**. It is interactive and collaborative, focusing on sharing our opinions and experiences. We don't have all the answers.

You will be working in groups to develop and share answers to these questions. You will all get as much out of the session as you put into it!

Agenda

10:00 - 10:25	Introduction - (*25 minutes*)
10:25 - 11:05	Workshop 1: The Content of the AD (*40 minutes*)
11:05 - 11:25	Group Session 1: share workshop 1 outputs (*20 minutes*)
11:25 - 11:45	Break (*20 minutes*)
11:45 - 12:25	Workshop 2: Tools and Techniques (*40 minutes*)
12:25 - 12:45	Group Session 2: share workshop 2 outputs (*20 minutes*)
12:45 - 13:00	Wrap-Up: Socialising your AD with stakeholders (*15 minutes*)

- Introduction
- Workshop 1: The Content of the AD
- Group Session 1: share workshop 1 outputs
- Break
- Workshop 2: Tools and Techniques
- Group Session 2: share workshop 2 outputs
- Wrap-Up: Socialising your AD with stakeholders

Introduction

The Architectural Description

An Architectural Description (AD) is a set of products which documents an architecture in a way which is understandable by its stakeholders, and demonstrates that the architecture has met their concerns.

- The products in an AD include views, models, principles, constraints etc. (as we will discuss during this workshop)
- The AD is also sometimes used to capture, consolidate and even refine other relevant information which has been defined elsewhere, such as business drivers, scope or requirements overview
- The AD has at the same time to present the essence of the architecture and its detail
- In other words, it has to provide an overall picture, which summarizes the whole system, but also decompose into enough detail so that it can be validated and the described system can be built
- The AD is not often a single document, but a collection of related artefacts which collectively document the architecture of the system

The AD is the place where all architecturally significant information about the system under consideration is recorded

Stakeholders and Concerns

A **stakeholder** in a software architecture is a person, group or entity with an interest in or concerns about the realisation of the architecture.

- A stakeholder may be a single person (eg a user, a test manager) or may be a group or entity (eg the support team)
- In some cases proxy stakeholders are required (eg the company lawyer representing a regulator)
- Stakeholders may be technical (developers, systems administrators) or non-technical (sponsor, business users)

A **concern** about an architecture is a requirement, an objective, an intention, or an aspiration which a stakeholder has for that architecture

- Many concerns will be common amongst stakeholders, but some concerns will be distinct and will often conflict
  - for example, users will want the system to be highly-functional and very performant, whereas acquirers (project sponsors) will want it to be inexpensive and delivered quickly
- Resolving such conflicts in a way that leaves stakeholders satisfied can be a significant challenge

An architecture is created solely to meet stakeholder needs,
and documented in the AD primarily to explain it to the stakeholders

Types of Stakeholder

Acquirers

- Oversee the procurement of the system or product

Assessors

- Oversee the system's conformance to standards and legal regulation.

Communicators

- Explain the system to other stakeholders via its documentation and training materials

Developers

- Construct and deploy the system (or lead the teams who do)

Maintainers

- Manage the evolution of the system once it is operational

Support Staff

- Provide support to users for the product or system when it is running

System Administrators

- Run the system once it has been deployed

Testers

- Test the system to ensure that it is suitable for use

Users

- Define the system's functionality, and ultimately make use of it

Suppliers

- A special class of stakeholder, who builds and/or supplies the hardware, software or infrastructure on which the system will run

A good Architectural Description is one that effectively and consistently communicates the key aspects of the architecture to the appropriate stakeholders in a way they can understand

Typical Stakeholder Concerns

Acquirers

- strategic alignment, return on investment, costs, timescales and plans

Assessors

- testing and compliance

Communicators

- understanding benefits, rationale, motivation and implications

Developers

- designing, building and testing the system

Maintainers

- development documentation, instrumentation, debug capabilities, change management

Support Staff

- problem diagnosis and resolution

System Administrators

- system monitoring and management, business continuity, availability and resilience, scalability

Testers

- establishing requirements, defining tests, test coverage, test harnesses

Users

- scope, functionality, ease of use

Suppliers

- commercial and licensing issues, successful deployment of their products

If you don’t adequately address all of these concerns in your AD, then you may not be able to "sell" your architecture to your stakeholders, which then threatens the success of your project

Architectural Significance

A concern, problem, or system element is **architecturally significant** if it has a wide impact on the structure of the system, or on its important quality properties such as performance, scalability, security, reliability or evolvability.

- In general, anything which is architecturally significant should be addressed in the AD, and the AD should not address concerns and system elements which are not architecturally significant
- Whether something is architecturally significant or not is a very subjective decision, influenced by the skills and experience of the audience, the time available to produce the AD, project context, circumstances and risks etc.
- A concern, problem or system element could be architecturally significant in one project, but not in another
  - for example. the design and configuration of a scheduling component might be hugely significant in a real-time process control system (because if it does not work properly processes will not execute at the right time) but not significant in an enquiry system most of whose activity is invoked by users

Deciding what is architecturally significant, and what should be left for consideration elsewhere, is one of the most important responsibilities of the architect

The Challenges of Producing an Effective AD

different stakeholders need different things from the the AD

you have to explain “why” and “so what” as well as just “what”

some stakeholders are very knowledgeable, others aren’t

you never have enough time to fully document the architecture

how much detail should you put into the AD?

at what point does the AD become a design? Does that matter?

you have to leave some areas undefined or vaguely defined without losing credibility

you need a “sales and marketing” document to convince stakeholders of your architecture’s viability, fitness for purpose, and cost-effectiveness

the AD needs to capture design decisions and the rationale clearly without confusing readers with options

the AD needs to be sufficiently detailed to unequivocally answer all the important decisions

Views, Viewpoints and Perspectives

A **view** of a software architecture is a representation of one or more aspects of an architecture that illustrates how it addresses concerns held by stakeholders

- Views solve the problem that it is not possible to capture all the functional features and quality properties of a complex architecture in a single model
- Views provide separation of concerns but require significant effort to ensure consistency and cohesion

A **viewpoint** is a collection of patterns, templates and conventions for constructing a view

- many standard definitions of viewpoints exist, including Rozanski & Woods (Functional, Information, Concurrency, Development, Deployment and Operational)

A **perspective** is a collection of activities, tactics and guidelines that are used to ensure that a system exhibits a particular set of related quality properties

- you apply perspectives to a view to ensure that it meets its non-functional requirements
- perspectives are a concept unique to Rozanski & Woods although all architectural approaches recognise the need to modify views so that the system exhibits the right quality properties

Architectural views are the cornerstone of most architectural descriptions (and the concept is enshrined in a standard, IEEE 1471 aka ISO 42010)

Other AD Content

Models

- an abstract representation of some aspect of an architecture
- views typically comprise a number of different models

Drivers and Goals

- set the project context, why it is needed and what it is intended to achieve

Scope

- defines the boundaries of the architecture - what is in and what is out

Principles

- fundamental statements of belief, approach or intent that guide the definition of an architecture

Requirements

- functional requirements define what the delivered system must do
- quality properties (aka non-functional requirements) define how the system must behave or operate

Constraints

- a fact or assertion that limits architectural choices

Standards, Guidelines, Policies etc

- used to enforce common approaches to design, development or operation

Decisions and Rationale

- key architectural decisions and why they were made
- alternatives considered and rejected (with reasons)

- Introduction
- Workshop 1: The Content of the AD
- Group Session 1: share workshop 1 outputs
- Break
- Workshop 2: Tools and Techniques
- Group Session 2: share workshop 2 outputs
- Wrap-Up: Socialising your AD with stakeholders

Workshop 1: The Content of the AD

Workshop 1: Content of the AD

The first workshop focusses on what should and can go into the AD.

- Clearly this includes architectural views, models and other means of representing the architecture itself, but may also include other content such as, scope definition, current context or principles
- The goal is to review what sort of content should be included in what circumstances, and what can or should be left out
  - For example, including alternatives which you considered and rejected can confuse your readers, and can also stir up controversy which you want to avoid

Building a list of essential and possible contents of an AD based on previous experience will provide us with a **template** that can be applied on subsequent projects

Some Things to Think About

- What is the role of the AD and who is its audience?
- To what extent is the AD a “persuading” document as well as just an “explaining” document?
- How do you decide what is architecturally significant and therefore merits consideration in the AD?
- How do you deal with things that are excluded because they are not architecturally significant?
- To what extent should non-architectural material be repeated or refined in the AD? (eg scope definition, requirements, constraints, plans, descriptions of external systems)
- To what extent should the same material be described in different documents aimed at different stakeholders?
  - For example, a “simplified” version of the AD produced for management or business users, plus
    - A more detailed / technical version of the AD aimed at developers
- If this approach is taken, how can it be done efficiently and how can consistency be maintained?

Ideally, you would derive a generic AD template that can be used for any type of system. In practice the template you produce today will reflect the context of your organisation and the types of system you work on

Workshop 1: Content of the AD

objectives	Consider and agree what should go into a generic Architectural Description.
inputs	our AD template
key questions to answer	what should be included in every AD? what should be considered for inclusion depending on circumstances and context? what should be omitted, either because it is not relevant or because it is documented elsewhere? if time is limited, what parts of the AD are essential and what are less important?
format	break up into small teams and work collaboratively to come to a consensus
output	list of AD contents and their relative importance (essential, important, optional) some key exclusions
duration	40 minutes, plus 20 minutes to share outputs

- Introduction
- Workshop 1: The Content of the AD
- Group Session 1: share workshop 1 outputs
- Break
- Workshop 2: Tools and Techniques
- Group Session 2: share workshop 2 outputs
- Wrap-Up: Socialising your AD with stakeholders

Group Session 1: Workshop 1 Outputs

- Introduction
- Workshop 1: The Content of the AD
- Group Session 1: share workshop 1 outputs
- Break
- Workshop 2: Tools and Techniques
- Group Session 2: share workshop 2 outputs
- Wrap-Up: Socialising your AD with stakeholders

Break
20 minutes

- Introduction
- Workshop 1: The Content of the AD
- Group Session 1: share workshop 1 outputs
- Break
- Workshop 2: Tools and Techniques
- Group Session 2: share workshop 2 outputs
- Wrap-Up: Socialising your AD with stakeholders

Workshop 2: Tools and Techniques

Workshop 2:
Tools and Techniques

The second workshop looks at tools and techniques.

- Teams will pick their "top 3" types of AD content (artefacts) from the first workshop, and explore how they can be documented to capture an appropriate level of detail and communicate the right messages to the readership.
- While architectural tool support is still very limited, we will encourage the audience to share their experiences, good and bad, in using tools to help automate the production and ongoing maintenance of these types of content.

Tools and Techniques for Documenting Architectures

Information Capture Tools

for capturing raw information for later processing

- word-processed documents
- electronic forms
- pen and paper

Diagramming Tools

for producing formal / informal diagrams

may also store metadada

- PowerPoint
- Visio or other general-purpose drawing tool
- UML tools
- architecture-specific tools

Architecture Description Languages (ADLs)

formal notations for documenting architectures

- Aesop, UniCon, xADL etc (+ tools)

Information Repositories

shared repositories for storing and managing architectural information

- spreadsheets
- databases
- UML tools
- portals and wikis

Structured Documentation Tools

for managing / presenting information in a structured way

- content management tools

Information Sharing tools

for sharing your completed AD with stakeholders

- presentation tools (Powerpoint)
- information portals (Sharepoint, wikis)

Workshop 2:
Tools and Techniques

objectives	Share experiences, good and bad, of tools and techniques for documenting an architecture in an AD.
inputs	output of previous workshop
key questions to answer	what tools and techniques are effective at capturing, documenting and presenting information? when should you use these tools and techniques? are the tools worth the cost / effort involved or are simple word processing tools just as effective? what works when time / budget is limited?
format	break up into small teams and work collaboratively to come to a consensus
output	examples of how you can use tools and techniques to document and present some important architectural artefacts what works well / what to avoid
duration	40 minutes, plus 20 minutes to share outputs

- Introduction
- Workshop 1: The Content of the AD
- Group Session 1: share workshop 1 outputs
- Break
- Workshop 2: Tools and Techniques
- Group Session 2: share workshop 2 outputs
- Wrap-Up: Socialising your AD with stakeholders

Group Session 2: Workshop 2 Outputs

- Introduction
- Workshop 1: The Content of the AD
- Group Session 1: share workshop 1 outputs
- Break
- Workshop 2: Tools and Techniques
- Group Session 2: share workshop 2 outputs
- Wrap-Up: Socialising your AD with stakeholders

Wrap-Up: Socialising your AD

Wrap-Up: Socialising your AD with Stakeholders

- Socialising your AD means:
  - making your stakeholders aware of its existence
  - making sure they understand its content (to the level they need to)
  - making sure they understand the implications of the decisions and compromises you have had to make
  - making sure they buy into it and support it
- If you don’t do this, then you are unlikely to build a system which conforms to your architecture
- All the good work you have done will be wasted - and the project has a significantly lower chance of success
- There are a number of techniques for doing this
  - open discussion - what have you used that works?

Nick Rozanski
Andy Longshaw
Eoin Woods

Sold!

How to Describe, Explain and Justify your Architecture