

OVERVIEW Documenting Software Architectures

An architecture only helps if the people who need it know it, understand it and apply it correctly, so communicating it is as important as creating it. Documenting an architecture means choosing the relevant views (4+1, Siemens, SCR...) for the stakeholders' concerns, documenting each view (primary presentation, element catalog, context diagram, variability guide, background), and documenting what spans the views (roadmap, overview, mappings, rationale, directory), usually following a template such as RUP or SEI and keeping only the minimum information needed.

Three uses

Education (introducing people),
communication (among stakeholders),
analysis (assuring quality attributes).

Getting started

Select the views, define the stakeholders,
identify their concerns, choose what and how
to document (a template).

Documenting a view

Primary presentation, element catalog,
context diagram, variability guide,
architecture background.

Beyond the views

Documentation roadmap, view
documentation, system overview, mapping
between views, rationale, directory.

HOW THE LECTURE IS ORGANIZED

Slides 1–7 Why document the architecture, its uses, how to start

Slides 8–13 Choosing views (4+1, Siemens, Herzum & Sims, SCR); stakeholders and concerns

Slides 14–16 Templates (RUP, SEI) and the minimum necessary information

Slides 17–22 Documenting a view, and documentation beyond views

Slides 23–25 Documenting quality attributes and summary

EXAM FOCUS

Final 252 Q8 took two 1-mark questions from this chapter: name two documentation templates (RUP and SEI), and list the major necessary information in an architecture document (problem description, context, rationale, principles and guidelines, structure, views, quality attributes).

HOW TO READ THIS PAGE

Every original slide is shown unchanged. **Yellow marks** on a slide point to a numbered note: hover or tap a mark to jump to its note, and hover a note to see its marks. Questions tagged **Past exam** appeared on a real paper; **Prediction** questions are likely angles, not confirmed.



Software Design and Architecture

[Documenting Software Architectures] – Chapter 09

NOTES

- 1 The final chapter: **documenting** the architecture, one of the architect's key tasks from Chapter 2 and the fourth step of the architecture process.

SLIDE 2 / 25 Lecture Outlines

Lecture Outlines

- Why Document the architecture
- Documenting a view
- Documenting beyond the view
- Documenting Quality

NOTES

- 1 Why document at all.
- 2 How to document one view.
- 3 What to document beyond the individual views.
- 4 How to document quality attributes.

SLIDE 3 / 25 Documenting the architecture

Documenting the architecture

- A software system's architecture may be its **most crucial** determinant of success or failure.
- Without an adequate architecture that delivers required function as well as **quality attributes**, the project will fail.
- **Communicating an architecture** to its stakeholders is as important a job as creating it in the first place.

NOTES

- 1 A system's architecture may be its **most crucial** determinant of success or failure.
- 2 Without an adequate architecture that delivers the required functions **and quality attributes**, the project will fail.
- 3 **Communicating** the architecture to its stakeholders is as important as creating it in the first place.

Documenting the architecture

- Even the best architecture will be useless if the people who need it
 - ✓ do not know what it is;
 - ✓ cannot understand it well enough to use, build, or modify it;
 - ✓ misunderstand it and apply it incorrectly.

- Therefore, it is important to document the architecture design properly.

NOTES

- 1 Even the best architecture is useless if the people who need it **don't know what it is...**
- 2 ...**can't understand it** well enough to use, build or modify it...
- 3 ...or **misunderstand it** and apply it incorrectly.
- 4 So the architecture must be documented **properly**.

Documenting the architecture

- Documenting a software architecture is a matter of:
 - ✓ Choosing a set of relevant views of the architecture,
 - ✓ Documenting each of those views, and then
 - ✓ Documenting information that applies to more than one view or to the set of views as a whole.
- ✓ In other words:
documenting software architecture includes *a way to choose the most relevant views*, *standard templates for documenting a view and documenting the information beyond views*, and definitions of the templates' content.

NOTES

- 1 Documenting an architecture means three things. **Choose a set of relevant views** (Chapter 2: no single view is enough).
- 2 **Document each** of those views.
- 3 **Document information that applies to more than one view**, or to the set as a whole.
- 4 In other words: a way to choose the views, **standard templates** for documenting a view and what lies beyond the views, and definitions of what goes in each template section.

Documenting the architecture

➤ Three primary uses for architecture documentation:

- ✓ **Education** -- introducing people to the project
- ✓ **Communication** -- among stakeholders
- ✓ **Analysis** -- assuring quality attributes

NOTES

- 1 **Education:** introducing people (new team members, external analysts) to the project.
- 2 **Communication** among stakeholders: the documentation is the shared reference for discussions and decisions.
- 3 **Analysis:** assuring the quality attributes, e.g. as the input to an ATAM evaluation (Chapter 8).

EXAM ANGLE

Prediction

Q1

Which is NOT one of the three primary uses of architecture documentation?

- A. Marketing the product
- B. Education
- C. Communication among stakeholders
- D. Analysis

Answer: A. Marketing the product

Documenting the architecture

- **To start with documentation, you need to (the goal of documentation):**
 - ✓ **Select common views** of Software Architecture Design (SAD)
 - ✓ **Define stakeholders**
 - ✓ **Identify stakeholders' concerns**
 - ✓ **Define what and how to document (select templates)**

NOTES

- 1 To start, **select the common views** for the Software Architecture Document (SAD).
- 2 **Define the stakeholders** (slide 12).
- 3 **Identify the stakeholders' concerns** (slide 13).
- 4 **Define what and how to document**, i.e. select templates (slides 14–16).

Possible Architectural View approaches

➤ Kruchten's 4+1 views

- ✓ **Logical view:** supports behavioral requirements. Key abstractions, which are objects or object classes
- ✓ **Process view:** addresses concurrency and distribution. Maps threads to objects.
- ✓ **Development view:** organization of software modules, libraries, subsystems, units of development.
- ✓ **Physical view:** maps other elements onto processing and communication nodes.
- ✓ **"Plus one" view:** Maps the other views onto important use cases to show how they work.

NOTES

- 1 Several view sets exist. The first is **Kruchten's 4+1** (Chapter 2, Lecture 3).
- 2 **Logical** (behavioural requirements: key abstractions as objects or classes), **process** (concurrency and distribution: threads mapped to objects), **development** (modules, libraries, subsystems), **physical** (elements mapped onto processing and communication nodes).
- 3 The **"plus one"** view maps the other four onto important use cases to show how they work together.

Possible Architectural View approaches(cont.)

➤ **Siemens Four-Views:**

- ✓ Conceptual view
- ✓ Module interconnection view
- ✓ Execution view
- ✓ Code view

➤ **Herzum & Sims:**

- ✓ Technical architecture
- ✓ Application architecture
- ✓ Project management architecture
- ✓ Functional architecture

NOTES

- 1 **Siemens four views:** conceptual, module interconnection, execution and code.
- 2 **Herzum & Sims:** technical, application, project-management and functional architectures. Different view sets organise the same system for different purposes.

Possible Architectural View approaches (cont.)

➤ **Software Cost Reduction method:**

- ✓ **Module view:** shows modules as units of encapsulation; used to isolate changes and achieve modifiability
- ✓ **Process view:** shows processes and how they synchronize and communicate at run-time; used to achieve performance
- ✓ **Uses view:** shows programs and how they depend on each other; used to achieve incremental development and the ability to quickly field subsets

NOTES

- 1 The **Software Cost Reduction (SCR)** method chooses each view for the quality it supports.
- 2 **Module view**: modules as units of encapsulation, to isolate changes (modifiability). **Process view**: processes and how they synchronise and communicate at run time (performance).
- 3 **Uses view**: programs and how they depend on each other, to support incremental development and fielding subsets quickly.

SLIDE 11 / 25 Possible Architectural View approaches

Possible Architectural View approaches (cont.)

Planning a **view** set requires understanding the needs of the **stakeholders**, how they will use the prepared documentation, and the resources available.

We covered the **4+1 views** in this course since it is more commonly used

NOTES

- 1 Planning a view set requires understanding the **stakeholders' needs**, how they will use the documentation, and the resources available.
- 2 This course used the **4+1 views** because they are the most common.

Define stakeholders

➤ Sample stakeholders:

- ✓ Project Manager
- ✓ Member of Development Team
- ✓ Testers and Integrators
- ✓ Maintainers
- ✓ Product Line Application Builder
- ✓ Customer
- ✓ End User
- ✓ Analyst
- ✓ Infrastructure Support
- ✓ Current and Future Architect

NOTES

- 1 Sample stakeholders of the documentation: customer, project manager, end user, development-team members, analysts.
- 2 Testers and integrators, infrastructure support, maintainers.
- 3 Current and future architects, and product-line application builders. Each needs different information, which is why the views are chosen per stakeholder.

Identify stakeholders' concerns

- Building a table that records the stakeholders' concerns and associated view helps in prioritizing the selected view based on the needs (some stakeholders may have extra weight)

No.	Stakeholder	Concerns to address	Suitable view
1.			
2.			
3.			
4.			
5.			

NOTES

- 1 Build a table that records each **stakeholder**, their concerns, and the view that addresses them.
- 2 It helps **prioritise** the views by need (some stakeholders carry more weight), and it exposes gaps: a concern with no suitable view means the documentation can't answer it.
- 3 Example row: maintainer → "where do I change the tax rules?" → development view.

Define How to Document

- After defining identifying the views, stakeholders' concerns, and what is going to be documented, a **template** is usually selected to prepare the documentation
- The following is an example of the **Rational Unified Process** template:

1. Introduction
 1. Purpose
 2. Scope
 3. Definitions, Acronyms, and Abbreviations
 4. References
 5. Overview
2. Architectural Representation
3. Architectural Goals and Constraints
4. Use-Case View
 1. Use-Case Realizations
5. Logical View
 1. Overview
 2. Architecturally Significant Design Packages
6. Process View
7. Deployment View
8. Implementation View
 1. Overview
 2. Layers
9. Data View (optional)
10. Size and Performance
11. Quality

NOTES

- 1 After choosing views, stakeholders, concerns and content, a **template** is selected to prepare the documentation.
- 2 Example: the **Rational Unified Process (RUP)** SAD template.
- 3 Its sections: 1 Introduction (purpose, scope, definitions, references, overview); 2 Architectural representation; 3 Architectural goals and constraints; 4 Use-case view; 5 Logical view; 6 Process view; 7 Deployment view; 8 Implementation view; 9 Data view (optional); 10 Size and performance; 11 Quality. Sections 4–8 are the 4+1 views.

EXAM ANGLE

Past exam · Final 252 · Q8-4 · 1 mark

Q1

State two types of document templates that can be used to document the architecture.

Answer: **RUP** (Rational Unified Process) template: introduction, architectural representation, goals and constraints, use-case, logical, process, deployment, implementation and data views, size and performance, quality. **SEI** (Software Engineering Institute) template: documentation roadmap, stakeholder representation, viewpoint definitions, architecture background, views, directory.

Define How to Document

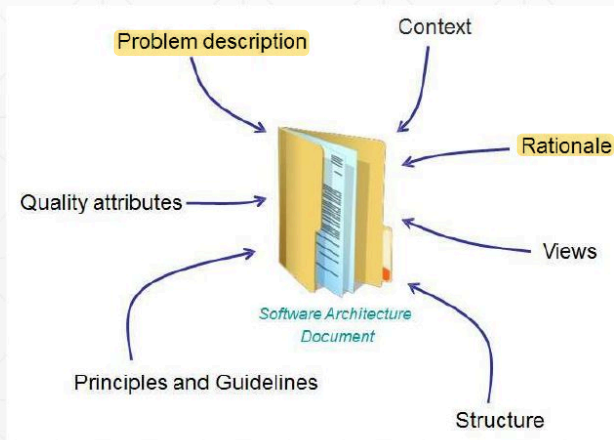
➤ The Software Engineering Institute (SEI) template:

1. Documentation Roadmap	1.5 Viewpoint Definitions
1. ...	1.5.1 <viewpoint> Viewpoint Definition
2. Stakeholder Representation	1.5.1.1 Abstract
3. Viewpoint Definitions	1.5.1.2 Stakeholders and Their Concerns Addressed
4. ...	1.5.1.3 Elements, Relations, Properties and Constraints
2. Architecture Background	1.5.1.4 Language(s) to Model/Represent Conforming Views
1. Problem Background	1.5.1.5 Applicable Evaluation/Analysis Techniques and Consistency/Completeness Criteria
1. System Overview	1.5.1.6 Viewpoint Source
2. Goals and Context	
3. Significant Driving Requirements	
2. Solution Background	
1. Architectural Approaches	3.1 <view name> View
2. Analysis Results	3.1.1 View Description
3. Requirements Coverage	3.1.2 View Packet Overview
4. Summary of Background Changes Reflected in Current Version	3.1.3 Architecture Background
3. Product Line Reuse Considerations	3.1.4 Variability Mechanisms
3. Views	3.1.5 View Packets
1. <Insert view name> View	3.1.5.1 View packet #1
4. ...	3.1.5.1.1 Primary Presentation
5. Directory	3.1.5.1.2 Element Catalog
1. Index	3.1.5.1.3 Context Diagram
2. Glossary	3.1.5.1.4 Variability Mechanisms
3. Acronym List	3.1.5.1.5 Architecture Background
	3.1.5.1.6 Related View Packets

NOTES

- 1 The **Software Engineering Institute (SEI)** template ("Views and Beyond").
- 2 Its sections: 1 **Documentation roadmap** (stakeholder representation, viewpoint definitions); 2 **Architecture background** (problem background, solution background with approaches, analysis results, requirements coverage); 3 **Views**, each documented as view packets with primary presentation, element catalog, context diagram, variability mechanisms and architecture background.
- 3 5 **Directory**: index, glossary, acronym list. It mirrors slides 17–22 almost section for section.

Define How to Document



- A software architecture document should have a **minimum information necessary** to understand the system but no more than that.
- Major necessary information are shown in the figure on the left

NOTES

- 1 A software architecture document should contain the **minimum information necessary** to understand the system, and no more. Every extra page is something to keep up to date.
- 2 The major necessary information (the figure): **problem description, context, rationale, views, structure, principles and guidelines, quality attributes.**

EXAM ANGLE

Past exam · Final 252 · Q8-5 · 1 mark

Q1

A software architecture document should have the minimum information necessary to understand the system but no more. What is the major necessary information to document?

Answer: Problem description · context · rationale · principles and guidelines · structure · views · quality attributes.

Documenting a View

1. A primary presentation

- ✓ Usually graphical (we call this a **cartoon**)
- ✓ May be textual -- e.g., a table
- ✓ If graphical, includes a **key explaining the notation** (or pointing to explanation)
- ✓ Shows elements and relationships among them
- ✓ Shows information you wish to convey about the view (view packet) first

➤ Many times, the primary presentation is all you get. However, it's most likely **not enough!**

NOTES

- 1 Documenting **one view**, part 1: the **primary presentation**, the main picture of the view.
- 2 It is usually graphical (informally a "cartoon"), but may be textual, such as a table.
- 3 If graphical, it includes a **key** explaining the notation, shows the elements and their relationships, and shows the most important information about the view first.
- 4 Often the primary presentation is all that gets written, but it is most likely **not enough**: boxes and lines without explanation are ambiguous. Hence parts 2–5.

Documenting a View (cont.)

2. An element catalog

- ✓ Explains the elements depicted in the primary presentation
- ✓ Lists elements and their properties (as defined by the relevant style guide)
- ✓ Explains relations, and any exceptions or additions to the relations shown in the primary presentation
- ✓ Interfaces of elements

3. A context diagram

- ✓ Shows how system (or portion shown in this view packet) relates to its environment.

NOTES

- 1 Part 2, the **element catalog**: explains each element in the primary presentation, lists elements and their properties, explains the relations (and any exceptions or additions to those shown), and gives the elements' **interfaces**.
- 2 Part 3, the **context diagram**: shows how the system (or the part shown in this view) relates to its environment: users, external systems, hardware.

Documenting a View (cont.)

4. Variability guide:

- ✓ Shows how to exercise any variation points that are a part of the architecture shown in this view.
- ✓ In some architectures, decisions are left **unbound until a later stage** of the development process, and yet the architecture must still be documented.

5. Architecture background

- ✓ **Rationale for design decisions** that apply to the entire view (or to that portion of the view being shown), including rejected alternatives and factors that constrained the design
- ✓ Analysis results validating the design decisions
- ✓ Assumptions about the environment and about the need that the system is fulfilling

NOTES

- 1 Part 4, the **variability guide**: shows how to use the variation points in this part of the architecture. Some decisions are deliberately left **unbound until later** (a configurable database, optional modules), yet the architecture must still be documented.
- 2 Part 5, the **architecture background**: the **rationale** for the design decisions in this view, including rejected alternatives and the factors that constrained the design; analysis results validating the decisions; and assumptions about the environment and the need the system fulfils.

EXAM ANGLE

Prediction

Q1

List the parts used to document one architectural view.

Answer: (1) Primary presentation (usually a diagram with a key); (2) element catalog (elements, properties, relations, interfaces); (3) context diagram; (4) variability guide; (5) architecture background (rationale, analysis results, assumptions).

Prediction

Q2

The part of a view's documentation that explains the elements shown in the diagram is the:

- A. Element catalog
- B. Variability guide
- C. Context diagram
- D. Documentation roadmap

Answer: A. Element catalog

Documentation beyond views

1. Documentation roadmap

- ✓ How the documentation is organized to serve a stakeholder
- ✓ List of views, with the elements/relations of each, and a statement of what the view is for
- ✓ Scenarios for using the documentation, showing which parts should be consulted

2. View Documentation

- ✓ Explanation of how each view is documented
- ✓ The standard organization for each view

NOTES

- 1 Beyond the views, part 1, the **documentation roadmap**: how the documentation is organised to serve each stakeholder, the list of views (their elements, relations and purpose), and scenarios for using the documentation ("to find X, read sections Y and Z").
- 2 Part 2, **view documentation**: how each view is documented, i.e. the standard organisation every view follows.

Documentation beyond views (cont.)

3. System overview

- ✓ Short prose description of the system's function, its users, and any important background or constraints.
- ✓ Provides your readers with a consistent mental model of the system and its purpose.

4. Mapping between views

- ✓ Establishes useful/insightful correspondence between various views
- ✓ View-to-view associations can be captured as tables

NOTES

- 1 Part 3, the **system overview**: a short prose description of the system's function, users, and important background or constraints, giving every reader the same mental model of the system and its purpose.
- 2 Part 4, the **mapping between views**: correspondences between elements in different views (this logical component runs as that process on that node), often as tables. Without mappings, views can quietly contradict each other.

Documentation beyond views (cont.)

5. **Rationale:**

- ✓ Documents the architectural decisions that apply to **more than one view**.
 - Documentation of background or organizational constraints or major requirements that led to decisions of system-wide import.
 - Decisions about which fundamental architecture patterns are used.

6. **Directory.**

- ✓ Set of reference material that helps readers find more information quickly.
 - Index of terms
 - **Glossary**
 - Acronym list

NOTES

- 1 Part 5, the **rationale**: the architectural decisions that apply to **more than one view**: the background, organisational constraints or major requirements behind system-wide decisions, and the choice of the fundamental architecture patterns.
- 2 Part 6, the **directory**: reference material to find information quickly: an index of terms, a **glossary** and an acronym list.

EXAM ANGLE

Prediction

Q1

List the six kinds of documentation that go beyond the individual views.

Answer: Documentation roadmap; view documentation (how each view is organised); system overview; mapping between views; rationale (cross-view decisions); directory (index, glossary, acronym list).

Documenting Quality Attributes

- The Architecture document should include a **rationale that explains the choice** of design approach and a discussion about the quality attribute requirements and tradeoffs.
- Architectural elements providing a service often have **quality attribute bounds** assigned to them
- Architecture documentation often contains a **mapping to requirements** that shows how requirements (including quality attribute requirements) are satisfied.
- Every quality attribute requirement will have a **constituency of stakeholders** who want to know that it is going to be satisfied. Quality attributes should be clearly identified in the document.

NOTES

- 1 The document should include a **rationale** explaining the choice of design approach and discussing the quality-attribute requirements and their trade-offs.
- 2 Architectural elements that provide a service often have **quality-attribute bounds** assigned to them (Chapter 2's quality tags: " ≤ 100 ms per order").
- 3 The documentation often contains a **mapping to requirements** showing how each requirement, including quality requirements, is satisfied.
- 4 Every quality requirement has a **constituency of stakeholders** who want to know it will be met, so quality attributes must be clearly identified in the document.

Summary

- You must understand the **uses to which the writing is to be put** and the **audience** for the writing.
- Architectural **documentation** serves as a means for **communication among** various stakeholders, not only up the management chain and down to the developers but also across to peers.
- An architecture is a complicated artifact, best expressed by focusing on **views**.
- You must choose the views to document and must choose a set of views that is both **minimal and adequate**.
- You must document not only the structure of the architecture but also the **behavior**.

NOTES

- 1 Understand the **uses** of the documentation and its **audience** before writing.
- 2 Architecture documentation is a means of **communication** among stakeholders: up to management, down to developers, and across to peers.
- 3 An architecture is complicated and is best expressed through **views**.
- 4 Choose a set of views that is both **minimal and adequate**.
- 5 Document the **behaviour** as well as the structure (e.g. sequence diagrams, not only component diagrams).

Lecture Outlines

- Why Document the architecture
- Documenting a view
- Documenting beyond the view
- Documenting Quality

NOTES

- 1 The outline again as a recap: why document, documenting a view, documenting beyond the views, documenting quality. This is the last chapter of SE322.

Original slides from the SE322 lecture deck, with study notes. Check exact wording against the [original deck](#). Notes and questions are study aids. Made by Shoug Alomran.