Documenting Software Architectures

Outline

- Introduction
- Uses of Architectural Documentation
- Views
- Documenting architectures

Introduction

- The software architecture plays a central role in system development.
- It is a blueprint for both the system and the project developing it.
- It defines work assignments.
- It is the primary carrier of system qualities.
- It is the conceptual glue that holds every phase of the project together for all of its many stakeholders.

Uses of Architectural Documentation

- A perfect architecture is useless if no one understands it or if key stakeholders misunderstand it
- Documentation is a crucial part of producing a good architecture
- Document should be
  - sufficiently abstract to be quickly understood by new employees
  - sufficiently detailed to serve as a blueprint for analysis

Uses of Architectural Documentation

- Architecture documentation is both prescriptive and descriptive.
- It satisfies the different needs of different stakeholders.
- It probably consists of many different pieces, each oriented to a specific set of stakeholders.
- It helps to educate people new to the project.

Stakeholders Who Use the Architectural Documentation

- Architect and requirements engineers who represent the customer(s)
- Architect and designers of constituent parts
- Implementors
- Testers and integrators
- Maintainers
- Designers of other systems with which this one must interoperate
- Quality attribute specialists
- Managers

Views

- A view is a *representation* of a coherent set of architectural elements, as written by and read by system stakeholders.
  - Views are not architectures
  - Views convey architectures

Views

- There are many views for an architecture.
- Documenting an architecture is a matter of documenting the relevant views and then adding documentation that applies to more than one view.
- The principle for documenting an architecture
  - Choosing the relevant views
  - Documenting a view
  - Documenting information that applies to more than one view

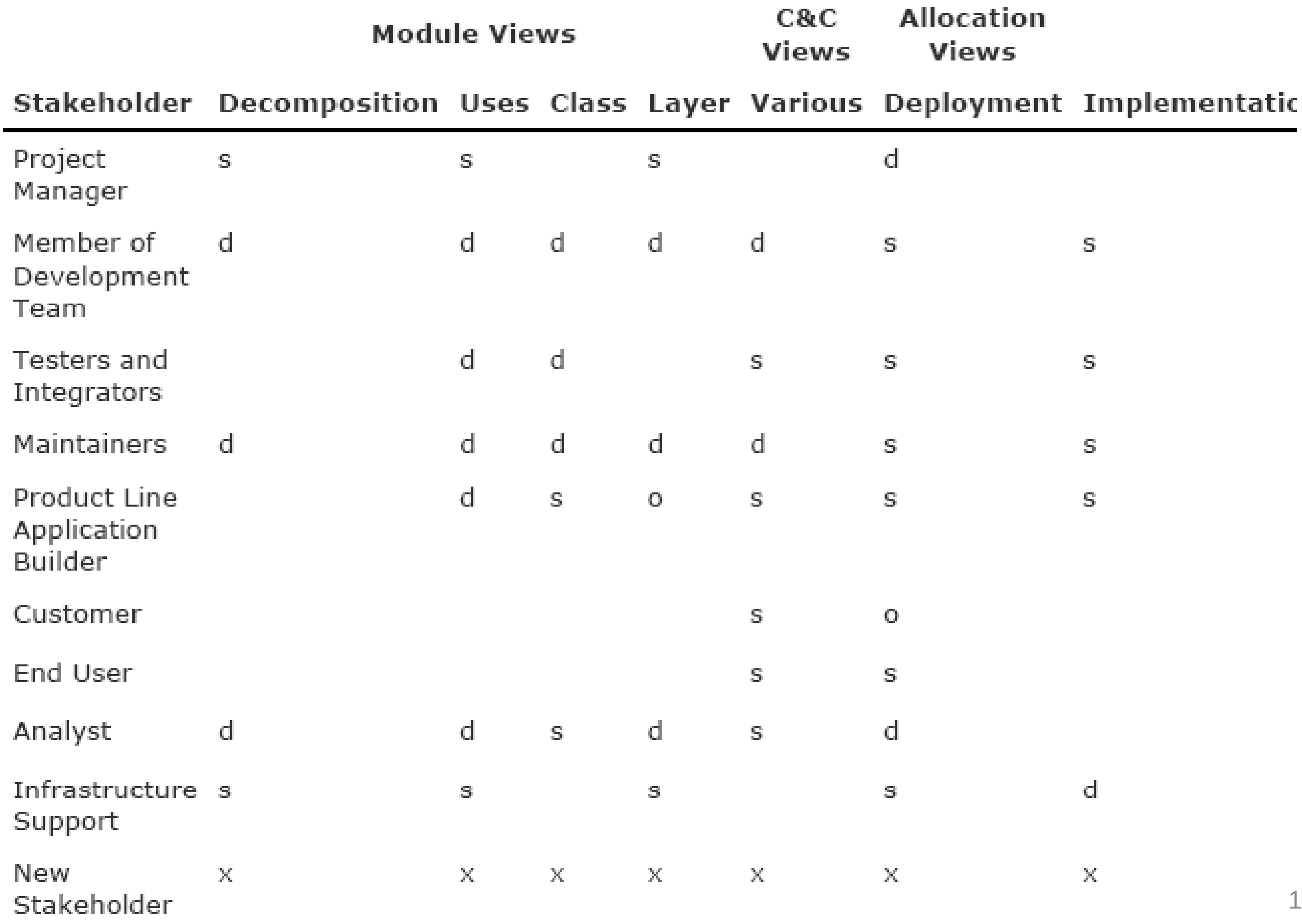
Choosing the Relevant Views

- What does “relevant” mean?

Choosing the Relevant Views

- Architects need to think about their software in at least three ways:
  - How it is structured as a set of implementation units (module view)
  - How it is structured as a set of elements that have runtime behavior and interactions (component-and-connector view)
  - How it relates to non-software structures in its environment (allocation view)

Stakeholders and Architecture documentation



	Module Views				C&C Views	Allocation Views	
Stakeholder	**Decomposition**	**Uses**	**Class**	**Layer**	**Various**	**Deployment**	**Implementatic**
Project Manager	s	s		s		d	
Member of Development Team	d	d	d	d	d	s	s
Testers and Integrators		d	d		s	s	s
Maintainers	d	d	d	d	d	s	s
Product Line Application Builder		d	s	o	s	s	s
Customer					s	o	
End User					s	s	
Analyst	d	d	s	d	s	d	
Infrastructure Support	s	s		s		s	d
New Stakeholder	x	x	x	x	x	x	x

Approach to Choosing Views

- Produce a candidate view list

  similar to previous table

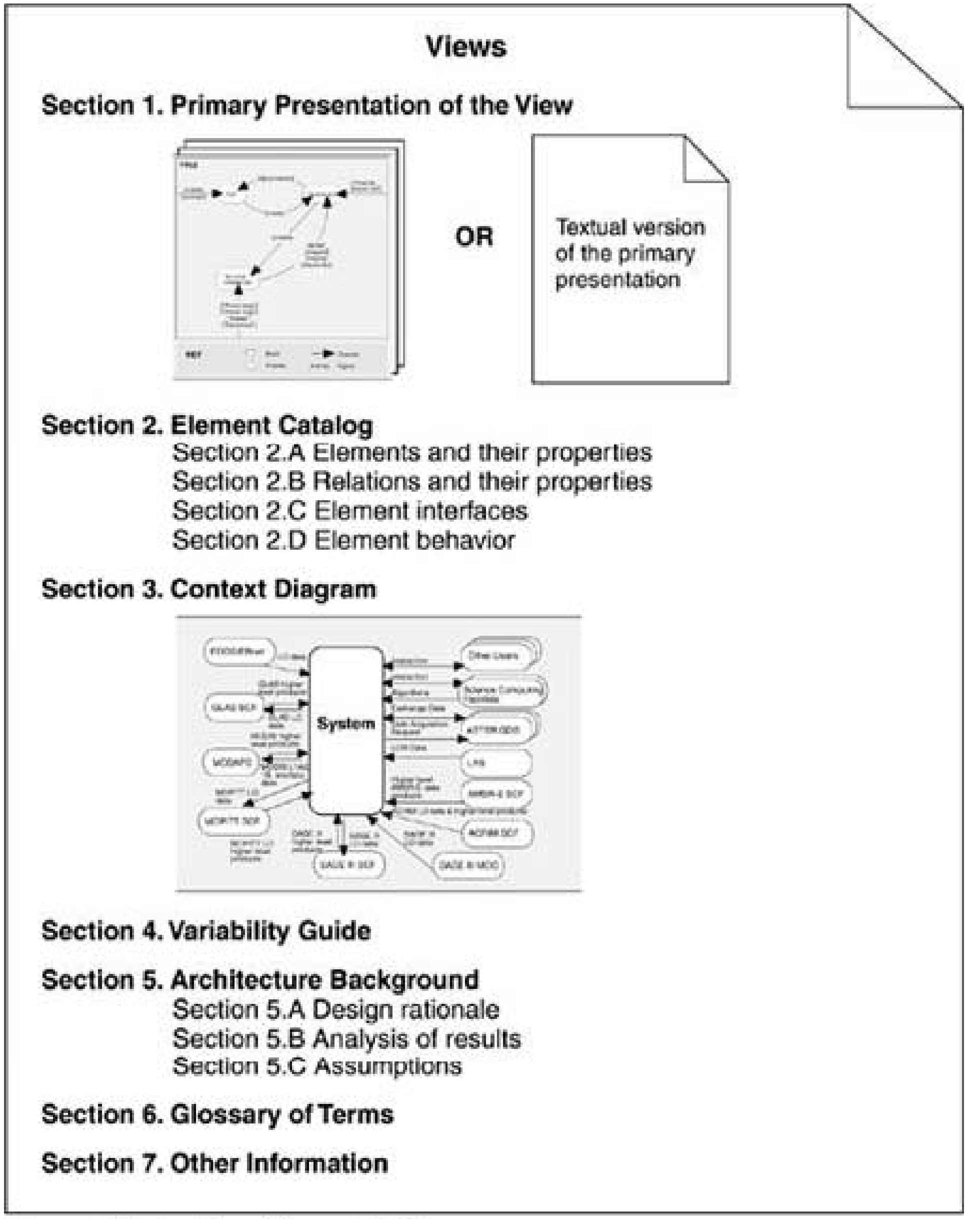
- Combine views

  reduce the size of the table

- Prioritize

  choose views to document first

Documenting a View



Views

Section 1. Primary Presentation of the View

OR

Textual version of the primary presentation

Section 2. Element Catalog

Section 2.A Elements and their properties
Section 2.B Relations and their properties
Section 2.C Element interfaces
Section 2.D Element behavior

Section 3. Context Diagram

Section 4. Variability Guide

Section 5. Architecture Background

Section 5.A Design rationale
Section 5.B Analysis of results
Section 5.C Assumptions

Section 6. Glossary of Terms

Section 7. Other Information

Source: Adapted from [Clements 03].

Documenting a View

- Primary presentation – this is usually graphical
- Element catalog – details those elements depicted in the primary presentation
- Context diagram – shows how the system relates to its environment
- Variability guide – shows how to exercise any variability points

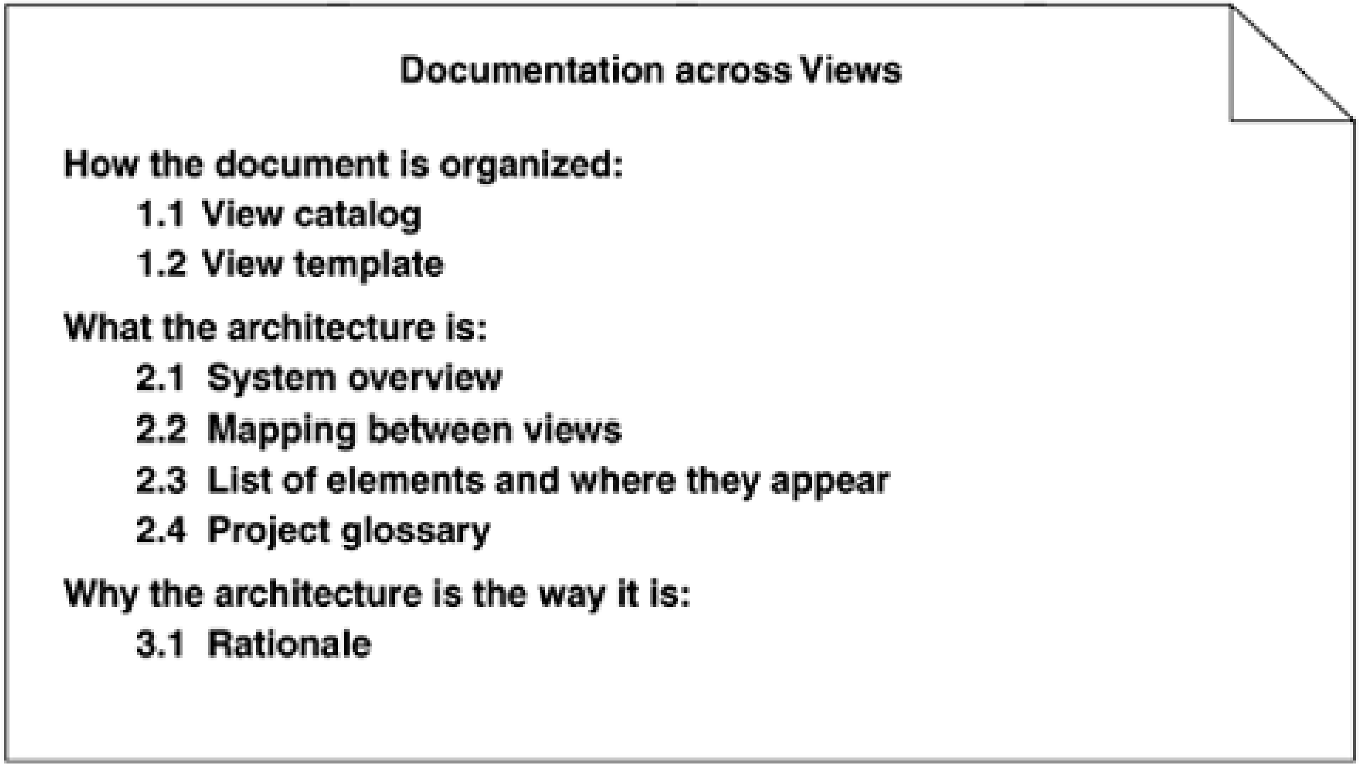
Documenting a View

- Architecture background – explains why this design came to be
- Glossary of terms – contains a brief description of terms used in the view
- Other information – contents of this section vary with the product and the organization

Documentation across views

- Cross-view documentation consists of just three major aspects
  - How the documentation is laid out and organized so that a stakeholder of the architecture can find the information he or she needs efficiently and reliably.
  - What the architecture is.
  - Why the architecture is the way it is

Documentation across views



Documentation across Views

How the document is organized:

- 1.1 View catalog
- 1.2 View template

What the architecture is:

- 2.1 System overview
- 2.2 Mapping between views
- 2.3 List of elements and where they appear
- 2.4 Project glossary

Why the architecture is the way it is:

- 3.1 Rationale

Source: Adapted from [Clements 03].