Software Architecture Documentation

Capturing the Right Stakeholders and Concerns

A Comprehensive Guide to Stakeholder Analysis,
Concern Identification, and Architecture Review

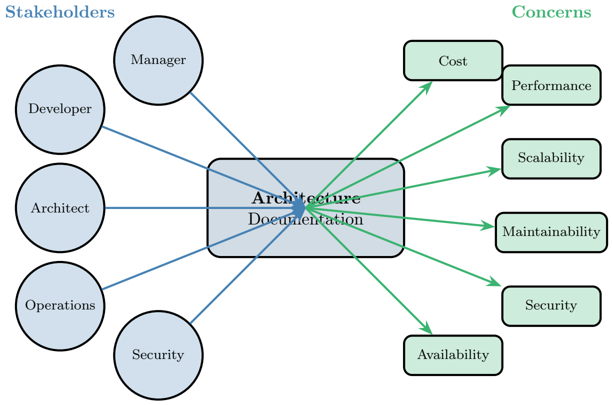
Architecture Documentation Series

Based on ISO/IEC/IEEE 42010, SEI Methods, and Industry Best Practices

December 9, 2025

Abstract

Effective architecture documentation requires understanding who will use it and what information they need. Stakeholder analysis and concern identification form the foundation of architecture documentation—they determine which views to create, what level of detail to provide, and how to organize information for maximum utility. This comprehensive guide establishes systematic processes for identifying stakeholders, eliciting and categorizing their concerns, mapping concerns to architectural views, and validating coverage through structured reviews. The document provides stakeholder taxonomies, concern catalogs, elicitation techniques, coverage matrices, and review question sets aligned with ISO/IEC/IEEE 42010 and SEI's Views and Beyond approach. Whether creating new architecture documentation or evaluating existing documentation, this guide ensures that the right stakeholders receive the right information to make informed decisions.



Contents

1 Introduction

1.1 Purpose of This Guide

This guide provides a systematic approach to identifying stakeholders and their concerns for software architecture documentation. It serves as both an instructional reference and a practical toolkit for architects creating or reviewing architecture documentation.

Definition

Stakeholder: An individual, team, organization, or class thereof, having an interest in a system (ISO/IEC/IEEE 42010). Stakeholders include anyone who uses, develops, operates, maintains, sponsors, or is affected by the system or its documentation.

Definition

Concern: An interest in a system relevant to one or more of its stakeholders (ISO/IEC/IEEE 42010). Concerns include system qualities, behaviors, constraints, and any other aspect of interest to stakeholders.

1.2 Why Stakeholders and Concerns Matter

Architecture documentation exists to serve stakeholders. Without understanding who will use the documentation and what information they need, architects risk creating documentation that is either incomplete (missing critical information) or bloated (containing unused detail).

Warning

Common Failures from Poor Stakeholder Analysis:
Missing stakeholders: Security team not consulted; security mechanisms inadequately documented; vulnerabilities discovered late.
Overlooked concerns: Operations team's deployment concerns ignored; documentation lacks deployment view; painful production rollouts.
Wrong level of detail: Executives receive highly technical views; developers receive high-level summaries; neither group served.
Undocumented assumptions: Stakeholder assumptions not captured; implicit requirements violated; costly rework.

1.3 Relationship to Architecture Documentation

Stakeholder and concern analysis drives all aspects of architecture documentation:

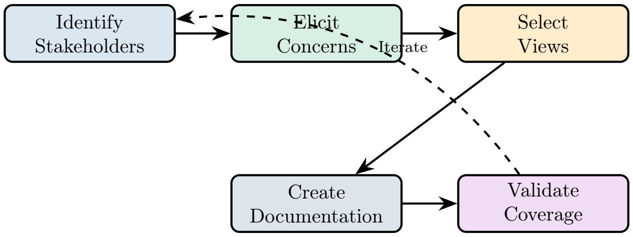


Figure 1: Stakeholder-Driven Documentation Process

1.4 Standards and Frameworks

This guide aligns with established standards and methodologies:

- **ISO/IEC/IEEE 42010:2011** — Systems and software engineering—Architecture description; defines stakeholders and concerns as fundamental concepts
- **SEI Views and Beyond** — Provides stakeholder-driven approach to view selection
- **TOGAF** — Includes stakeholder management in Architecture Development Method
- **Rozanski & Woods** — Software Systems Architecture with stakeholder-concern-viewpoint framework
- **IEEE 1471-2000** — Predecessor to ISO 42010; established stakeholder-centric approach

2 Stakeholder Taxonomy

2.1 Stakeholder Categories

Stakeholders can be organized into categories based on their relationship to the system.

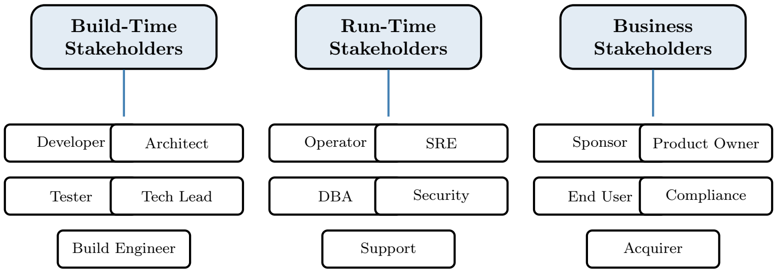


Figure 2: Stakeholder Categories

2.2 Comprehensive Stakeholder Catalog

The following catalog provides a reference list of common stakeholders. Not all stakeholders apply to every project; use this as a checklist to ensure no relevant stakeholders are overlooked.

Table 1: Build-Time Stakeholders

Stakeholder	**Role Description**	**Primary Interests**
Software Architect	Defines system structure and key decisions	Technical coherence; quality attributes; design rationale
Developer	Implements system components	Component responsibilities; interfaces; dependencies; coding standards
Tech Lead	Leads development team; makes technical decisions	Team boundaries; integration points; technical feasibility
QA Engineer	Designs and executes tests	Testability; component boundaries; test environments
Build Engineer	Manages build and CI/CD pipelines	Build dependencies; artifact structure; deployment automation
Database Admin	Designs and maintains databases	Data models; performance; backup/recovery; migrations
UI/UX Designer	Creates user interface designs	User interaction patterns; frontend architecture
Technical Writer	Creates documentation	Documentation structure; terminology; audience needs
Security Engineer	Ensures security requirements met	Security mechanisms; threat models; compliance
Performance Engineer	Optimizes system performance	Performance characteristics; bottlenecks; benchmarks

Table 2: Run-Time Stakeholders

Stakeholder	**Role Description**	**Primary Interests**
System Administrator	Deploys and configures systems	Deployment procedures; configuration; administration tools
Operator	Monitors and maintains running systems	Operational procedures; monitoring; troubleshooting
SRE	Ensures reliability and availability	Reliability patterns; incident response; SLOs
Network Engineer	Manages network infrastructure	Network topology; protocols; security zones
Help Desk / Support	Provides user support	Error messages; diagnostic procedures; escalation paths
End User	Uses the system directly	Functionality; usability; response time
System Integrator	Integrates with other systems	APIs; protocols; data formats; integration patterns
Data Analyst	Analyzes system data	Data access; reporting capabilities; analytics

Stakeholder	Role Description	Primary Interests
Auditor	Verifies compliance	Audit trails; access controls; compliance evidence

Table 3: Business Stakeholders

Stakeholder	Role Description	Primary Interests
Project Sponsor	Funds and champions the project	ROI; schedule; budget; risk
Product Owner	Defines product requirements	Feature delivery; business value; roadmap
Project Manager	Manages project execution	Schedule; resources; dependencies; risks
Business Analyst	Translates business needs	Requirements; use cases; business rules
Enterprise Architect	Ensures enterprise alignment	Standards compliance; reuse; integration
Compliance Officer	Ensures regulatory compliance	Regulations; policies; audit requirements
Legal Counsel	Manages legal aspects	Licensing; liability; contracts; IP
Procurement	Acquires products and services	Vendor evaluation; contracts; costs
Customer	External purchaser/user	Capabilities; quality; support; pricing
Partner	External organization with integration	APIs; SLAs; data sharing; security

2.3 Stakeholder Identification Process

Best Practice

Systematic Stakeholder Identification:

1. **Start with the catalog:** Review the stakeholder catalog above as a checklist
2. **Analyze the lifecycle:** Consider who is involved in each phase:
   - Requirements and design
   - Development and testing
   - Deployment and operations
   - Maintenance and evolution
   - Retirement and replacement
3. **Follow the money:** Identify who funds, purchases, or benefits financially
4. **Follow the data:** Identify who creates, uses, or is affected by system data
5. **Consider external entities:** Regulators, partners, competitors, public
6. **Ask stakeholders:** Each identified stakeholder may know of others
7. **Review organizational charts:** Identify affected departments and roles
8. **Document and validate:** Create stakeholder list; validate with project leadership

2.4 Stakeholder Prioritization

Not all stakeholders have equal influence or interest. Prioritization helps focus effort on the most important stakeholders.

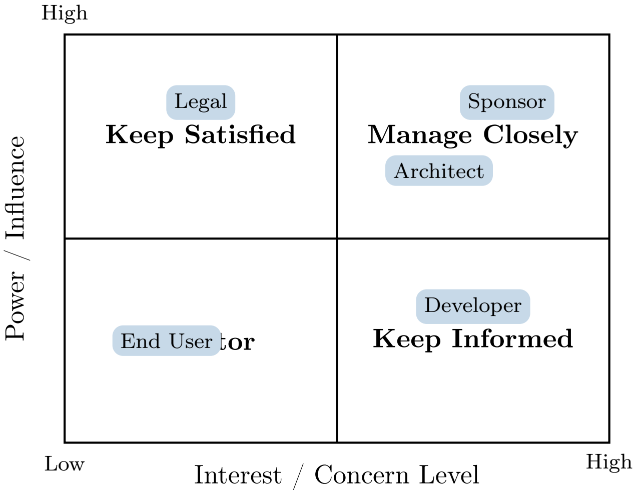


Figure 3: Stakeholder Power/Interest Matrix

Table 4: Stakeholder Engagement Strategies

Quadrant	Characteristics	Engagement Strategy
Manage Closely	High power, high interest	Active engagement; regular consultation; address concerns promptly
Keep Satisfied	High power, low interest	Keep informed of major decisions; don't overwhelm with detail
Keep Informed	Low power, high interest	Regular updates; address concerns; leverage their knowledge
Monitor	Low power, low interest	Minimal effort; periodic check-ins; monitor for changing interest

3 Concern Taxonomy

3.1 What is a Concern?

Definition

A **concern** is any interest in a system that is relevant to one or more stakeholders. Concerns encompass system qualities (performance, security), system behaviors (functionality), constraints (technology mandates, regulations), and any other aspect that stakeholders care about.

Concerns drive architecture documentation by determining what information must be captured and communicated.

3.2 Concern Categories

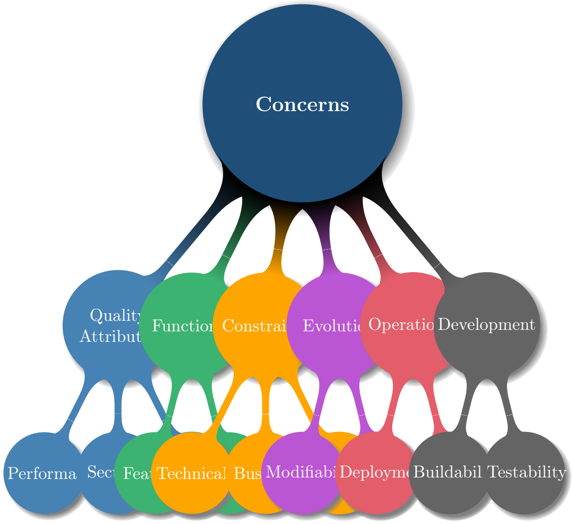


Figure 4: Concern Category Mind Map

3.3 Comprehensive Concern Catalog

3.3.1 Quality Attribute Concerns

Quality attributes (also called non-functional requirements) are often the most architecturally significant concerns.

Table 5: Quality Attribute Concerns

Quality Attribute	Definition	Typical Stakeholders
Performance	Response time, throughput, resource utilization	End users; Operations; Performance engineers
Scalability	Ability to handle increased load	Architects; Operations; Capacity planners
Availability	System uptime; fault tolerance	End users; Operations; SREs; Business
Reliability	Consistency of correct behavior over time	End users; QA; Operations
Security	Protection against threats; confidentiality; integrity	Security team; Compliance; End users
Maintainability	Ease of modification and correction	Developers; Architects; Maintainers
Modifiability	Ability to make changes at reasonable cost	Architects; Developers; Product owners
Testability	Ease of testing and verification	QA engineers; Developers
Usability	Ease of use; user experience	End users; UX designers; Product owners

Quality Attribute	Definition	Typical Stakeholders
Interoperability	Ability to work with other systems	System integrators; Partners; Architects
Portability	Ability to run in different environments	Operations; Architects; Enterprise architects
Deployability	Ease of deployment and updates	Operations; DevOps; Release managers
Monitorability	Ability to observe system behavior	Operations; SREs; Support
Recoverability	Ability to recover from failures	Operations; Business continuity; DBAs
Cost	Development and operational expenses	Sponsors; Finance; Management

3.3.2 Functional Concerns

Table 6: Functional Concerns

Concern	Description	Typical Stakeholders
Feature completeness	All required features implemented	Product owners; Business analysts
Data integrity	Data remains accurate and consistent	DBAs; Data engineers; Compliance
Business rules	Correct implementation of business logic	Business analysts; Domain experts
User workflows	Support for user tasks and processes	End users; UX designers
Integration	Connectivity with external systems	System integrators; Partners
Reporting	Generation of required reports	Business users; Data analysts

3.3.3 Constraint Concerns

Table 7: Constraint Concerns

Concern	Description	Typical Stakeholders
Technology mandates	Required technologies or platforms	Enterprise architects; IT governance
Standards compliance	Adherence to technical standards	Architects; Compliance
Regulatory compliance	Meeting legal/regulatory requirements	Compliance; Legal; Auditors

Budget constraints	Development and operational budget limits	Sponsors; Finance; Management
Schedule constraints	Delivery timeline requirements	Project managers; Sponsors
Resource constraints	Available people and skills	Project managers; HR
Legacy integration	Compatibility with existing systems	Enterprise architects; Operations
Vendor relationships	Use of specific vendors or products	Procurement; Enterprise architects

3.3.4 Development Concerns

Table 8: Development Concerns

Concern	**Description**	**Typical Stakeholders**
Code organization	How code is structured and organized	Developers; Tech leads; Architects
Build process	How system is compiled and packaged	Build engineers; DevOps
Development environment	Tools and setup for development	Developers; Tech leads
Team structure	How teams are organized around code	Management; Tech leads
Coding standards	Conventions and style guidelines	Developers; Tech leads
Dependency management	External library and service dependencies	Developers; Security; Architects
Version control	Source code management approach	Developers; Build engineers

3.3.5 Operational Concerns

Table 9: Operational Concerns

Concern	**Description**	**Typical Stakeholders**
Deployment	How system is deployed to production	Operations; DevOps; SREs
Configuration	How system is configured per environment	Operations; Developers
Monitoring	How system behavior is observed	Operations; SREs
Logging	What is logged and how	Operations; Security; Support

Alerting	How problems are detected and reported	Operations; SREs
Backup/Recovery	Data protection and restoration	DBAs; Operations; Business continuity
Disaster recovery	Recovery from major failures	Business continuity; Operations
Capacity planning	Planning for growth	Capacity planners; Operations
Incident response	Handling production issues	SREs; Support; Operations

4 Concern Elicitation

4.1 Elicitation Techniques

Multiple techniques should be used to ensure comprehensive concern identification.

Table 10: Concern Elicitation Techniques

Technique	**Description**	**When to Use**
Stakeholder Interviews	One-on-one discussions with stakeholders	Key stakeholders; sensitive concerns; deep exploration
Workshops	Facilitated group sessions	Multiple stakeholders; cross-cutting concerns; consensus building
Questionnaires	Written questions distributed to stakeholders	Large stakeholder groups; remote stakeholders; initial gathering
Document Analysis	Review of existing documentation	Legacy systems; regulatory requirements; organizational standards
Catalog Review	Walk through concern catalog with stakeholders	Ensuring completeness; prompting discussion
Scenario Analysis	Explore system behavior in specific situations	Quality attributes; edge cases; failure modes
Prototype Review	Stakeholders review early system versions	User-facing concerns; usability; workflow validation
Quality Attribute Workshop (QAW)	SEI method for eliciting quality requirements	Quality attribute prioritization; scenario development

4.2 Interview Guidelines

Stakeholder Interview Template

Pre-Interview Preparation

- Review stakeholder's role and organizational context
- Identify likely concerns based on role
- Prepare role-specific questions
- Schedule 60-90 minutes

Opening (5 minutes)

- Introduce purpose: understanding their perspective on the system
- Explain how information will be used
- Confirm time available

Core Questions (45-60 minutes)

1. Describe your role and how you interact with this system
2. What does success look like for you with this system?
3. What are your biggest concerns or worries about the system?
4. What information about the system do you need to do your job?
5. What problems have you experienced with similar systems?
6. What quality attributes matter most to you? (performance, security, etc.)
7. Are there constraints or regulations that affect your work?
8. Who else should we talk to about concerns like yours?
9. What questions do you have that the architecture documentation should answer?

Closing (5-10 minutes)

- Summarize key concerns captured
- Ask if anything was missed
- Explain next steps and follow-up
- Thank stakeholder for their time

Post-Interview

- Document concerns within 24 hours
- Categorize concerns using taxonomy
- Identify concerns requiring follow-up
- Send summary to stakeholder for validation

4.3 Quality Attribute Workshop (QAW)

The SEI's Quality Attribute Workshop is a structured method for eliciting and prioritizing quality attribute concerns.

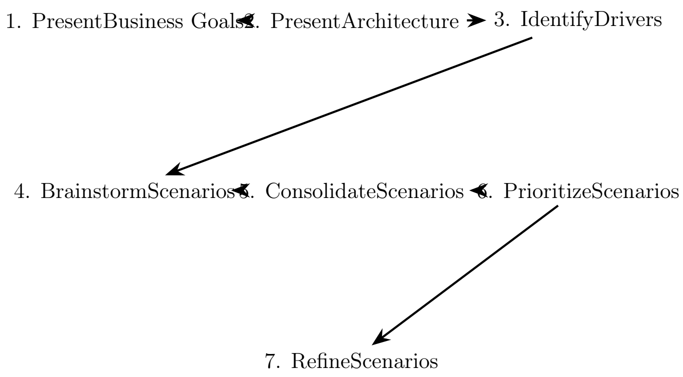


Figure 5: Quality Attribute Workshop Process

QAW Outputs:

- Prioritized list of quality attribute scenarios
- Business goals and their architectural implications
- Architectural drivers (most important quality concerns)
- Stakeholder consensus on priorities

5 Mapping Concerns to Views

5.1 The Mapping Challenge

Once concerns are identified, they must be mapped to architectural views that address them. This mapping ensures that:

- Every concern is addressed by at least one view
- Views are created for the most important concerns
- Stakeholders know where to find information about their concerns

5.2 Concern-View Mapping Matrix

Table 11: Concern-View Mapping Matrix

Concern	**Module**	**C&C**	**Deploy**	**Data**	**Context**	**Rationale**
Performance	○	●	●	○		○
Scalability		●	●			○
Availability		●	●			○
Security	○	●	●	●	○	●
Maintainability	●					○
Modifiability	●	○				●
Testability	●	○				
Deployability			●			
Code organization	●					
Data integrity				●		

Concern	Module	C&C	Deploy	Data	Context	Rationale
Integration		●			●	
Cost			●			●

● = Primary view ○ = Secondary view

5.3 Stakeholder-View Navigation Guide

Table 12: Stakeholder-View Navigation

Stakeholder	Primary Views	Key Information Sought
Developer	Module; C&C	Component responsibilities; interfaces; dependencies
Architect	All views	Design decisions; trade-offs; quality attributes
Tech Lead	Module; C&C	Team boundaries; integration points
QA Engineer	Module; C&C	Testable components; interfaces; scenarios
Operations	Deployment; C&C	Infrastructure; monitoring; procedures
SRE	Deployment; C&C	Reliability patterns; failure modes; recovery
Security	C&C; Deployment; Data	Security mechanisms; trust boundaries
DBA	Data; Deployment	Data models; storage; backup
Product Owner	Context; Rationale	Capabilities; decisions; roadmap implications
Sponsor	Context; Rationale	Business value; risks; costs

6 Review Question Sets

This section provides structured question sets for reviewing stakeholder and concern coverage in architecture documentation.

6.1 Question Set Overview

Key Point

Purpose: These question sets verify that the architect has identified appropriate stakeholders and concerns and that these are adequately represented in the architecture description. They enable both self-review by architects and formal reviews by stakeholder groups.

6.2 Questions for All Stakeholders

Stakeholder Self-Assessment Questions

Instructions: Each stakeholder should answer these questions independently, then discuss findings in a review session.

Role and Concern Identification

1. State your stakeholder role(s). What decisions do you make or actions do you take that depend on the architecture?
2. List all concerns you have that pertain to the architecture. For each concern:
   - Why does this concern matter to you?
   - What would be the impact if this concern were not addressed?
   - How would you measure whether the concern is satisfied?
3. Prioritize your concerns (High/Medium/Low) based on their impact on your ability to perform your role.

Coverage Assessment

4. Find and record all places in the architecture description where your stakeholder role is explicitly mentioned or addressed.
5. For each of your concerns, find and record where in the documentation that concern is:
   - Listed or acknowledged
   - Actually addressed with architectural content
6. Rate how well each concern is addressed:
   - **Well addressed:** Clear, complete information enabling decisions
   - **Partially addressed:** Some information, but gaps remain
   - **Acknowledged only:** Listed but not substantively addressed
   - **Not addressed:** No mention or coverage
7. For concerns not well addressed, describe:
   - What specific information is missing?
   - What decisions can you not make without this information?
   - What is the impact of this gap on project success?

Gap Identification

8. Identify any stakeholder roles you are aware of that are not represented in the architect's stakeholder list. For each:
   - What is the role?
   - Why should they be included?
   - What concerns would they likely have?
9. Identify any concerns that should be in the documentation but are not, including:
   - Concerns for missing stakeholders
   - Additional concerns for your own role
   - Cross-cutting concerns affecting multiple stakeholders
10. For each missing concern, assess:
    - Criticality (High/Medium/Low)
    - Impact if not addressed
    - Suggested approach for addressing

Usability Assessment

11. Rate the ease of finding information relevant to your concerns:
    - Easy: Intuitive navigation; information where expected
    - Moderate: Required some searching; eventually found
    - Difficult: Significant effort to locate; scattered information

6.3 Questions for the Architect

Architect Self-Assessment Questions

Stakeholder Identification Process

1. Describe the process you used to identify stakeholders:
   - What sources did you consult? (org charts, project docs, interviews)
   - What techniques did you use? (workshops, interviews, checklists)
   - How did you validate completeness?
2. For each stakeholder category (build-time, run-time, business), explain why specific roles were included or excluded.
3. How did you prioritize stakeholders for documentation focus?

Concern Elicitation Process

4. Describe how you elicited concerns from stakeholders:
   - Which stakeholders did you interview or consult?
   - What elicitation techniques did you use?
   - How did you ensure concerns were accurately captured?
5. How did you prioritize concerns for documentation focus?
6. For quality attribute concerns, describe how you developed quality attribute scenarios or requirements.

Coverage Demonstration

7. Show where in the documentation each identified stakeholder role is addressed.
8. For each documented concern, show:
   - Which view(s) address the concern
   - How the concern is satisfied by the architecture
   - Any trade-offs made affecting the concern
9. Demonstrate that every concern traces to at least one stakeholder.
10. Demonstrate that every view traces to at least one concern.

Framework Alignment

11. If using an architectural framework (TOGAF, ISO 42010, etc.):
    - Show where framework-required stakeholders are addressed
    - Show where framework-required concerns are addressed
    - Explain any deviations from framework guidance

6.4 Review Session Guidelines

Template

Stakeholder Review Session Agenda

Pre-Session (1-2 weeks before)
- Distribute architecture documentation to all reviewers
- Distribute question sets with response templates
- Request completed responses 2 days before session
- Compile and analyze responses; identify themes

Session Agenda (Half-day workshop)
1. Opening (15 minutes)
- Welcome and introductions
- Review session objectives and ground rules
- Overview of architecture documentation structure

2. Stakeholder Coverage Review (45 minutes)
- Present compiled stakeholder coverage findings
- Discuss missing or underrepresented stakeholders
- Reach consensus on stakeholder list completeness

3. Concern Coverage Review (90 minutes)
- Present compiled concern coverage findings
- Walk through each concern category
- Identify gaps and inadequately addressed concerns
- Prioritize gaps by impact

4. Architect Response (30 minutes)
- Architect responds to findings
- Clarifications and additional context
- Identify quick fixes vs. significant updates

5. Action Planning (30 minutes)
- Document required changes
- Assign owners and deadlines
- Schedule follow-up review if needed

6. Closing (15 minutes)
- Summarize decisions and actions
- Gather feedback on review process
- Thank participants

7 Coverage Assessment

7.1 Stakeholder Coverage Matrix

Table 13: Stakeholder Coverage Assessment Template

Stakeholder	Listed	Concerns Identified	Views Address	Coverage	Gaps
Developer	✓	✓	Module, C&C	Full	–
Architect	✓	✓	All	Full	–
Operations	✓	✓	Deploy, C&C	Partial	Runbooks missing
Security	✓	✓	Security View	Full	–
Product Owner	✓	Partial	Context	Partial	Roadmap impact
DBA	–	–	–	None	Not consulted

Coverage: Full = All concerns addressed; Partial = Some gaps; None = Not covered

7.2 Concern Traceability Matrix

Table 14: Concern Traceability Matrix

Concern	Stakeholders	Addressed In	Priority	Status
Performance	Users, Ops, Perf Eng	C&C View, Deploy View	High	Addressed
Security	Security, Compliance	Security View, Rationale	High	Addressed
Scalability	Architect, Ops	C&C View, Deploy View	High	Addressed
Maintainability	Developer, Tech Lead	Module View	Medium	Addressed
Testability	QA, Developer	Module View	Medium	Partial
Deployability	Ops, DevOps	Deploy View	Medium	Addressed
Cost	Sponsor, Finance	Rationale	High	Partial
Data migration	DBA, Ops	–	Medium	Not addressed

7.3 Gap Analysis Template

Template

Documentation Gap Report

Gap ID: GAP-001
Type: Missing Stakeholder / Missing Concern / Inadequate Coverage
Description: [Clear description of what is missing or inadequate]
Affected Stakeholders: [Who is impacted by this gap]
Impact Assessment:

- **Severity:** High / Medium / Low
- **Project Impact:** [How this affects project success]
- **Decision Impact:** [What decisions cannot be made]

Recommended Resolution:

- **Action:** [What should be done]
- **Owner:** [Who is responsible]
- **Effort:** [Estimated effort to resolve]
- **Deadline:** [When it should be completed]

Status: Open / In Progress / Resolved / Deferred
Resolution Notes: [How the gap was addressed]

8 Governance and Maintenance

8.1 Stakeholder Registry

Maintain a living registry of stakeholders that evolves with the project.

Table 15: Stakeholder Registry Template

Name	Role	Category	Priority	Contact	Status
J. Smith	Lead Developer	Build-time	High	jsmith@	Active
A. Jones	SRE Lead	Run-time	High	ajones@	Active
B. Wilson	Product Owner	Business	High	bwilson@	Active
C. Brown	Security Arch	Build-time	Medium	cbrown@	Active
D. Lee	Project Sponsor	Business	High	dlee@	Active

8.2 Review Cadence

Table 16: Stakeholder Review Cadence

Review Type	Frequency	Purpose
Stakeholder List Review	Quarterly	Verify stakeholder list current; add/remove as needed
Concern Coverage Review	Per major release	Verify all concerns still addressed
New Stakeholder Onboarding	As needed	Validate documentation meets new stakeholder needs
Full Documentation Review	Annually	Comprehensive review of all coverage

8.3 Change Management

Best Practice

When to Update Stakeholder/Concern Analysis:

- New stakeholder roles identified or organizational changes occur
- New concerns emerge from project experience
- Regulatory or compliance requirements change
- Technology changes introduce new concerns
- Post-incident reviews reveal documentation gaps
- Stakeholder feedback indicates unmet needs
- Architecture undergoes significant changes

9 Appendix A: Stakeholder Interview Questions by Role

9.1 Developer Interview Questions

1. What components are you responsible for developing?
2. What information do you need about other components to do your work?
3. What development standards or patterns should the architecture enforce?
4. What are your concerns about testability of the system?
5. What dependencies cause you the most problems?
6. What would make the codebase easier to work with?

9.2 Operations Interview Questions

1. What systems will you be responsible for operating?
2. What deployment procedures do you need documented?
3. What monitoring and alerting capabilities do you require?
4. What are your concerns about system reliability?
5. What backup and recovery procedures do you need?

6. What capacity planning information do you require?

9.3 Security Interview Questions

1. What security standards must the system comply with?
2. What are the most critical assets to protect?
3. What threat vectors concern you most?
4. What security controls do you expect to see documented?
5. What audit and compliance requirements exist?
6. What security testing will be required?

9.4 Business Stakeholder Interview Questions

1. What business goals does this system support?
2. What is the expected return on investment?
3. What are the critical success factors?
4. What risks concern you most?
5. What schedule or budget constraints exist?
6. How will success be measured?

10 Appendix B: Review Checklist

10.1 Stakeholder Coverage Checklist

- □ All stakeholder categories considered (build-time, run-time, business)
- □ Key stakeholders identified by name and role
- □ Stakeholder prioritization documented
- □ Each stakeholder's concerns explicitly listed
- □ Stakeholders validated the concern list
- □ Missing stakeholders identified and addressed
- □ Stakeholder contact information maintained

10.2 Concern Coverage Checklist

- □ All concern categories considered (quality, functional, constraint, etc.)
- □ Concerns prioritized by importance
- □ Each concern traced to stakeholders
- □ Each concern mapped to architectural view(s)
- □ Quality attribute scenarios developed for key concerns
- □ Trade-offs between concerns documented
- □ Missing concerns identified and addressed

10.3 Documentation Usability Checklist

- □ Stakeholder navigation guide provided
- □ Concern-to-view mapping documented

☐ Views organized for stakeholder consumption
☐ Terminology consistent and defined
☐ Cross-references enable navigation
☐ Search/index functionality available (if online)

11 Appendix C: Glossary

Architecture Description
A work product used to express an architecture (ISO 42010)

Concern An interest in a system relevant to stakeholders

Quality Attribute
A measurable or testable property of a system

Quality Attribute Scenario
A specific, measurable requirement for a quality attribute

Stakeholder An individual, team, or organization with interest in a system

Stakeholder Analysis
Process of identifying and understanding stakeholders

View A representation of a system from the perspective of related concerns

Viewpoint Specification of conventions for constructing and using a view

12 Appendix D: References

1. ISO/IEC/IEEE 42010:2011. *Systems and software engineering—Architecture description.*
2. Clements, P., et al. (2010). *Documenting Software Architectures: Views and Beyond* (2nd ed.). Addison-Wesley.
3. Rozanski, N., & Woods, E. (2011). *Software Systems Architecture* (2nd ed.). Addison-Wesley.
4. Bass, L., Clements, P., & Kazman, R. (2021). *Software Architecture in Practice* (4th ed.). Addison-Wesley.
5. Barbacci, M., et al. (2003). "Quality Attribute Workshops (QAWs), Third Edition." SEI Technical Report CMU/SEI-2003-TR-016.
6. The Open Group. (2018). *TOGAF Standard, Version 9.2.* Van Haren Publishing.
7. IEEE 1471-2000. *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems.*
8. Lattanze, A. (2008). *Architecting Software Intensive Systems: A Practitioner's Guide.* Auerbach Publications.