

# Applying the Cognitive Dimensions of API Usability to Improve API Documentation Planning

A method for describing the gap that API documentation must bridge



The cognitive dimensions characterize API design and API users to understand and describe the documentation that they require.

## 12 Cognitive Dimensions

Documentation impact hypotheses for the dimensions that depend on the API design

API Design Dependent

- **Consistency**  
An inconsistent interface will require more content overall than a consistent one.
- **Domain Correspondence**  
An API that corresponds to the domain will require less documentation than one that does not.
- **Role Expressiveness**  
An API with transparent role expressiveness will require less explanation than an opaque one.
- **Working Framework**  
An API with a local working framework will be easier to explain than one with a system Working Framework.
- **API Elaboration**  
APIs that require elaboration to use will require more documentation than those that don't.
- **API Viscosity**  
An API that is difficult to change will need more explanation than one that is easy to change.

Documentation impact hypotheses for the dimensions that depend on the audience and the API

Interaction Dependent

- **Abstraction Level**  
An API designed to a different abstraction level than the user prefers will require more explanation.
- **Learning Style**  
An API that supports a different learning style than the user prefers will require more accommodation.
- **Penetrability**  
An API that supports a different degree of penetrability than the user prefers will require more complex documentation.
- **Progressive Evaluation**  
An API that supports a different level of progressive evaluation than the user prefers will have more complex examples.
- **Work-Step Unit**  
Tutorials and examples will be easier to write when the audience's preferred Work-Step Unit matches the API's.
- **Premature Commitment**  
Tutorials will be easier to write when the audience's preferred Premature Commitment matches the API's.

## Developing the method

### Validate the method

Does it really identify documentation gaps?

Are all dimensions necessary?

1. Simplify dimensions and criteria
2. Collect, share, and review data
3. Validate documentation impacts
4. Improve and repeat

### Test in practice

How can it be used in the real world?

- Checklists or interviews
- Heuristic evaluation
- Quantitative scoring

For what can it be used in the real world?

- Informal awareness
- Formal planning
- Post-project review

### Next steps

Where to go from here?

- Interest groups
- Volunteers network
- Research partnerships
- Other opportunities?