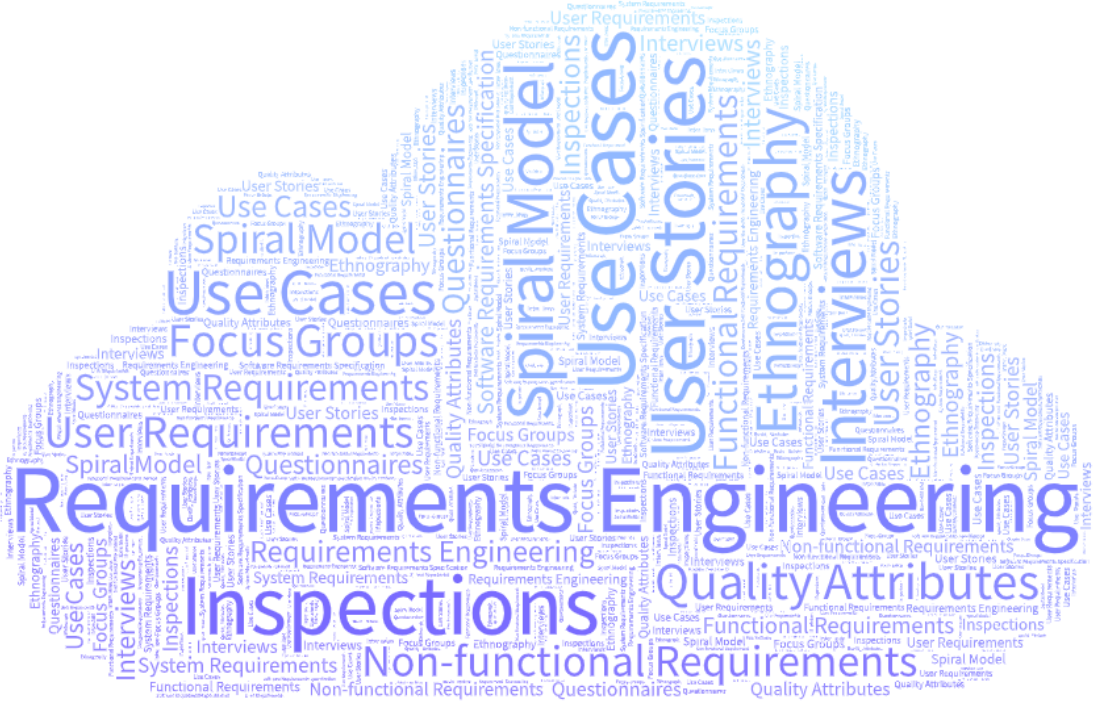
CS3213: Foundations of Software Engineering

Requirements Engineering Overview

Requirements Engineering (RE)

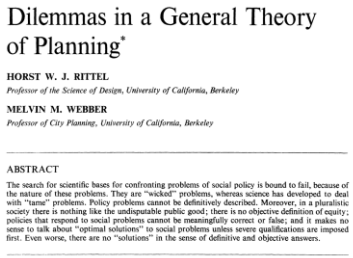


Word cloud generated using https://wordart.com/create

Image by Gemini

Image by Gemini

Wicked Problem



Dilemmas in a General Theory of Planning*

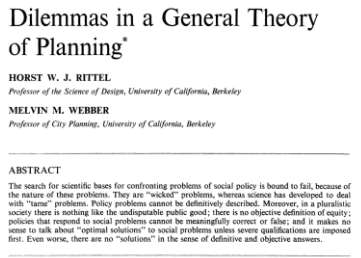
HORST W. J. RITTEL
Professor of the Science of Design, University of California, Berkeley
MELVIN M. WEBBER
Professor of City Planning, University of California, Berkeley

ABSTRACT

The search for scientific bases for confronting problems of social policy is bound to fail, because of the nature of these problems. They are "wicked" problems, whereas science has developed to deal with "tame" problems. Policy problems cannot be definitively described. Moreover, in a pluralistic society there is nothing like the undisputable public good; there is no objective definition of equity; policies that respond to social problems cannot be meaningfully correct or false; and it makes no sense to talk about "optimal solutions" to social problems unless severe qualifications are imposed first. Even worse, there are no "solutions" in the sense of definitive and objective answers.

- There is no definitive formulation of a wicked problem.
- Wicked problems have no stopping rule.
- Solutions to wicked problems are not true-or-false, but better or worse.
- There is no immediate and no ultimate test of a solution to a wicked problem.
- Every solution to a wicked problem is a "one-shot operation"; because there is no opportunity to learn by trial and error, every attempt counts significantly.
- Every wicked problem is essentially unique.
- Wicked problems do not have an enumerable (or an exhaustively describable) set of potential solutions, nor is there a well-described set of permissible operations that may be incorporated into the plan.
- Every wicked problem can be considered to be a symptom of another problem.
- The existence of a discrepancy representing a wicked problem can be explained in numerous ways. The choice of explanation determines the nature of the problem's resolution.
- The social planner has no right to be wrong (i.e., planners are liable for the consequences of the actions they generate).

Wicked Problem



Dilemmas in a General Theory of Planning*

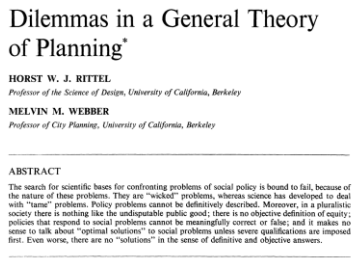
HORST W. J. RITTEL
Professor of the Science of Design, University of California, Berkeley
MELVIN M. WEBBER
Professor of City Planning, University of California, Berkeley

ABSTRACT

The search for scientific bases for confronting problems of social policy is bound to fail, because of the nature of these problems. They are "wicked" problems, whereas science has developed to deal with "tame" problems. Policy problems cannot be definitively described. Moreover, in a pluralistic society there is nothing like the undisputable public good; there is no objective definition of equity; policies that respond to social problems cannot be meaningfully correct or false; and it makes no sense to talk about "optimal solutions" to social problems unless severe qualifications are imposed first. Even worse, there are no "solutions" in the sense of definitive and objective answers.

- There is no definitive formulation of a wicked problem.
- Wicked problems have no stopping rule.
- Solutions to wicked problems are not true-or-false, but better or worse.
- There is no immediate and no ultimate test of a solution to a wicked problem.
- Every solution to a wicked problem is a "one-shot operation"; because there is no opportunity to learn by trial and error, every attempt counts significantly.
- Every wicked problem is essentially unique.
- Wicked problems do not have an enumerable (or an exhaustively describable) set of potential solutions, nor is there a well-described set of permissible operations that may be incorporated into the plan.
- Every wicked problem can be considered to be a symptom of another problem.
- The existence of a discrepancy representing a wicked problem can be explained in numerous ways. The choice of explanation determines the nature of the problem's resolution.
- The social planner has no right to be wrong (i.e., planners are liable for the consequences of the actions they generate).

Wicked Problem



Dilemmas in a General Theory of Planning*

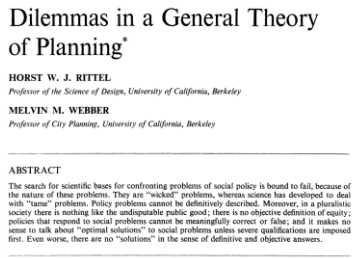
HORST W. J. RITTEL
Professor of the Science of Design, University of California, Berkeley
MELVIN M. WEBBER
Professor of City Planning, University of California, Berkeley

ABSTRACT

The search for scientific bases for confronting problems of social policy is bound to fail, because of the nature of these problems. They are "wicked" problems, whereas science has developed to deal with "tame" problems. Policy problems cannot be definitively described. Moreover, in a pluralistic society there is nothing like the undisputable public good; there is no objective definition of equity; policies that respond to social problems cannot be meaningfully correct or false; and it makes no sense to talk about "optimal solutions" to social problems unless severe qualifications are imposed first. Even worse, there are no "solutions" in the sense of definitive and objective answers.

- There is no definitive formulation of a wicked problem.
- Wicked problems have no stopping rule.
- Solutions to wicked problems are not true-or-false, but better or worse.
- There is no immediate and no ultimate test of a solution to a wicked problem.
- Every solution to a wicked problem is a "one-shot operation"; because there is no opportunity to learn by trial and error, every attempt counts significantly.
- Every wicked problem is essentially unique.
- Wicked problems do not have an enumerable (or an exhaustively describable) set of potential solutions, nor is there a well-described set of permissible operations that may be incorporated into the plan.
- Every wicked problem can be considered to be a symptom of another problem.
- The existence of a discrepancy representing a wicked problem can be explained in numerous ways. The choice of explanation determines the nature of the problem's resolution.
- The social planner has no right to be wrong (i.e., planners are liable for the consequences of the actions they generate).

Wicked Problem



Dilemmas in a General Theory of Planning*

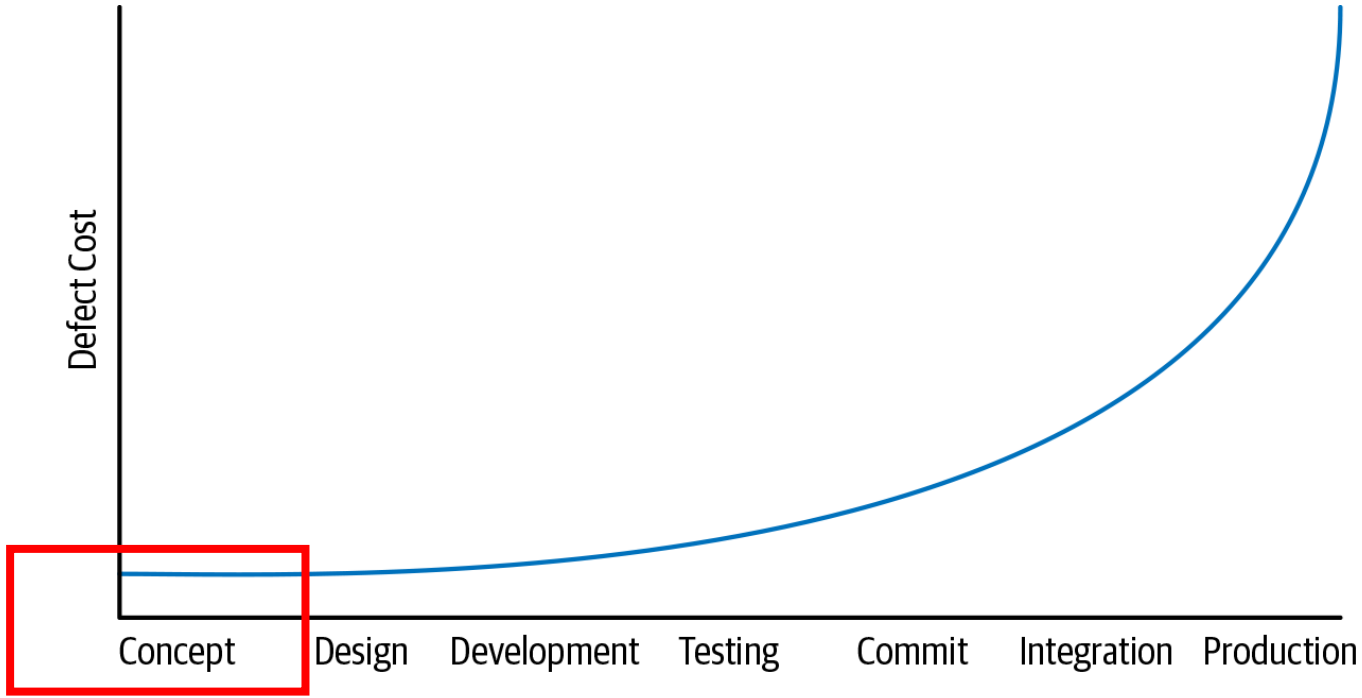
HORST W. J. RITTEL
Professor of the Science of Design, University of California, Berkeley
MELVIN M. WEBBER
Professor of City Planning, University of California, Berkeley

ABSTRACT

The search for scientific bases for confronting problems of social policy is bound to fail, because of the nature of these problems. They are "wicked" problems, whereas science has developed to deal with "tame" problems. Policy problems cannot be definitively described. Moreover, in a pluralistic society there is nothing like the undisputable public good; there is no objective definition of equity; policies that respond to social problems cannot be meaningfully correct or false; and it makes no sense to talk about "optimal solutions" to social problems unless severe qualifications are imposed first. Even worse, there are no "solutions" in the sense of definitive and objective answers.

- There is no definitive formulation of a wicked problem.
- Wicked problems have no stopping rule.
- Solutions to wicked problems are not true-or-false, but better or worse.
- There is no immediate and no ultimate test of a solution to a wicked problem.
- Every solution to a wicked problem is a "one-shot operation"; because there is no opportunity to learn by trial and error, every attempt counts significantly.
- Every wicked problem is essentially unique.
- Wicked problems do not have an enumerable (or an exhaustively describable) set of potential solutions, nor is there a well-described set of permissible operations that may be incorporated into the plan.
- Every wicked problem can be considered to be a symptom of another problem.
- The existence of a discrepancy representing a wicked problem can be explained in numerous ways. The choice of explanation determines the nature of the problem's resolution.
- The social planner has no right to be wrong (i.e., planners are liable for the consequences of the actions they generate).

Why RE?



https://abseil.io/resources/swe-book/html/ch01.html

Why RE?

Activity	Requirements as basis for …
Customer relation and communication	Communication between stakeholders
Software design	Structure and behavior of a software system
Quality assurance and acceptance	As a basis for testing and acceptance of final product
Maintenance and evolution	Dictate the quality of the system realization, risks, and overhead in operations

Modified based on: Daniel Mendez' Lecture slides on "Requirements Engineering"

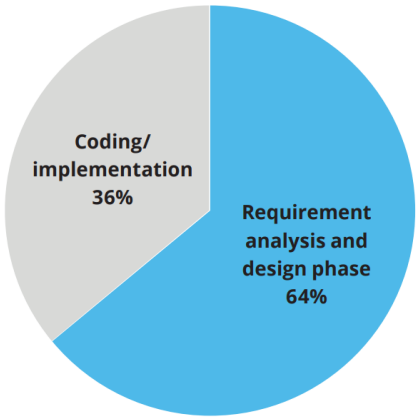
Why RE?

Your bad requirements are costing you money

Author: **Anjalie Rajkumar**
January 2022

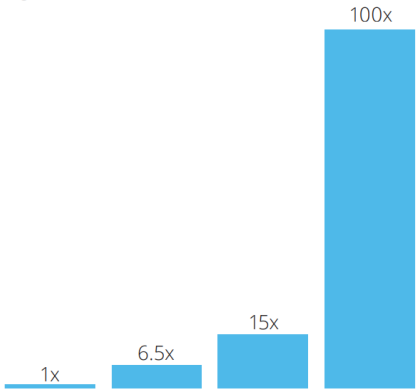
Origin of software defects
(Source: Crosstalk, the Journal of defence software engineering)

Figure 1



Relative costs to fix software defects
(Source: IBM systems sciences institute)

Figure 2



https://www2.deloitte.com/content/dam/Deloitte/uk/Documents/technology/deloitte-uk-your-bad-requirements-are-costing-you-money.pdf

Stakeholders

Stakeholder: a stakeholder is a person or organization who influences a system's requirements or who is impacted by that system.

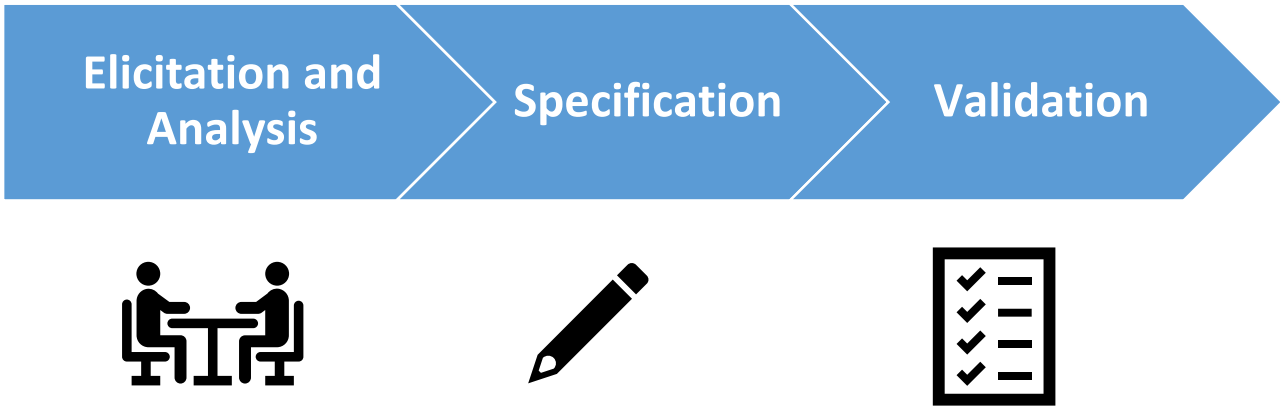
Image by Gemini

Identifying Stakeholders: Snowball Sampling

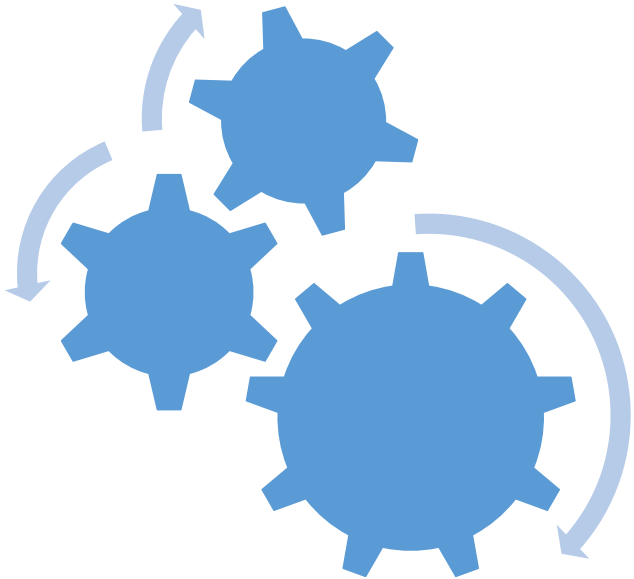
Image by Gemini

https://discovery.ucl.ac.uk/id/eprint/744/1/1.7_stake.pdf

Key Activities



Process Models



Process model

Functional Requirements

Functional: The functional requirements for a system describe what the system should do

Image by ChatGPT

Non-functional Requirements

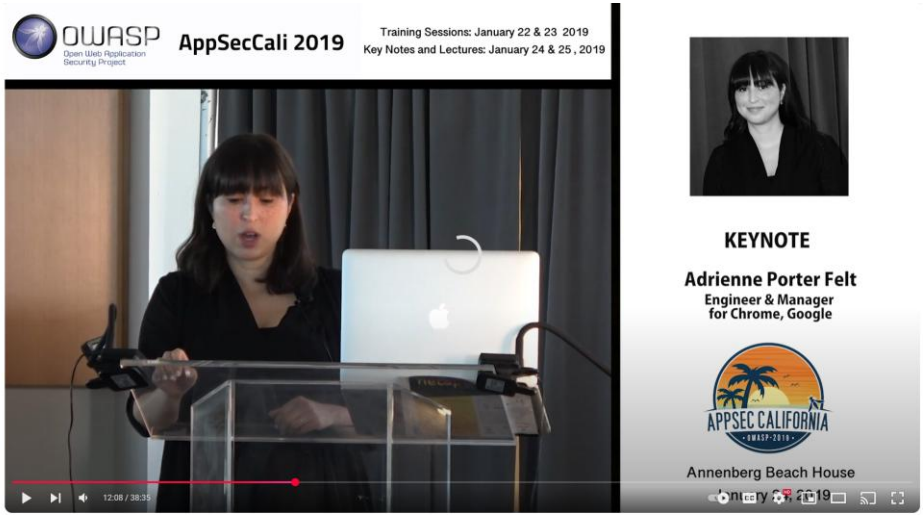
Non-functional: non-functional requirements are requirements that are not directly concerned with the specific services delivered by the system to its users

Conflicting Non-functional Requirements

Image by ChatGPT

Case Study: Chrome

Key goals/non-functional requirements of Google Chrome:
simple, fast, secure, and stable



AppSecCali 2019 Keynote - How to Slay a Dragon - Adrienne Porter Felt

https://www.youtube.com/watch?v=nF81_IU6tHI

NDSS 2022 MADWeb - Shipping security at scale in the Chrome browser - Adriana Porter Felt

https://www.youtube.com/watch?v=eNmrCAq1GR4

Case Study: Chrome

- Compression proxy
  - Prioritized security over performance
- Removals of features threatening security
  - Prioritized stability over security
- Site isolation
  - Prioritized security over performance

Quality Attributes vs. Constraints

NON-FUNCTIONAL REQUIREMENTS

QUALITY ATTRIBUTES
– The "Ilities" –

- Scalability
- Reliability
- Usability

CONSTRAINTS
– The "Musts" –

- Android compatible
- PDPA compliant
- Written in Python

Image by Gemini

Internal vs. External Quality Attributes

External quality	Brief description
Availability	The extent to which the system's services are available when and where they are needed
Installability	How easy it is to correctly install, uninstall, and reinstall the application
Integrity	The extent to which the system protects against data inaccuracy and loss
Interoperability	How easily the system can interconnect and exchange data with other systems or components
Performance	How quickly and predictably the system responds to user inputs or other events
Reliability	How long the system runs before experiencing a failure
Robustness	How well the system responds to unexpected operating conditions
Safety	How well the system protects against injury or damage
Security	How well the system protects against unauthorized access to the application and its data
Usability	How easy it is for people to learn, remember, and use the system
Internal quality	**Brief description**
Efficiency	How efficiently the system uses computer resources
Modifiability	How easy it is to maintain, change, enhance, and restructure the system
Portability	How easily the system can be made to work in other operating environments
Reusability	To what extent components can be used in other systems
Scalability	How easily the system can grow to handle more users, transactions, servers, or other extensions
Verifiability	How readily developers and testers can confirm that the software was implemented correctly

Software Requirements. Third Edition. Wiegers and Beatty.

Quality Attributes

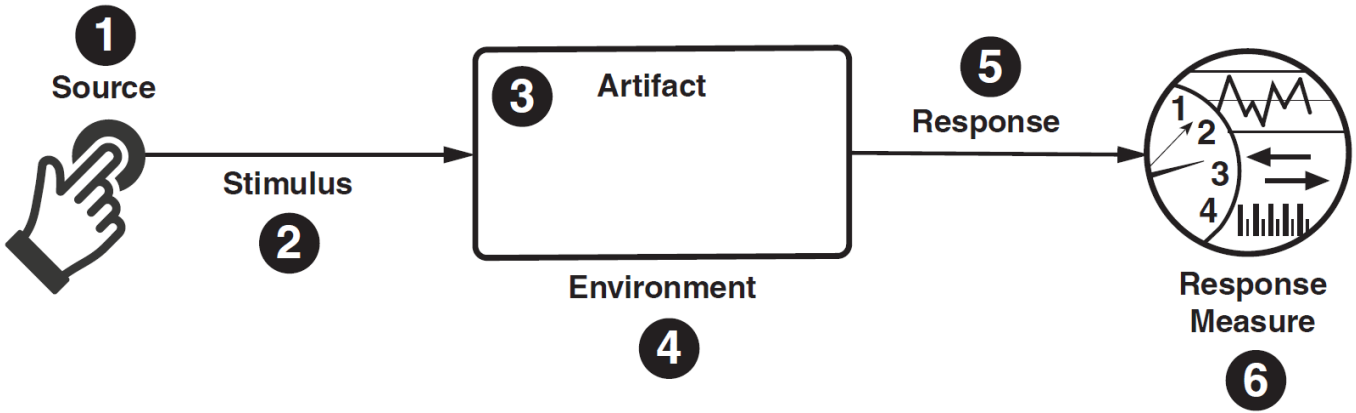
The system should be highly available.

https://tenor.com/de/view/no-god-please-no-gif-21601753

Quality Attribute

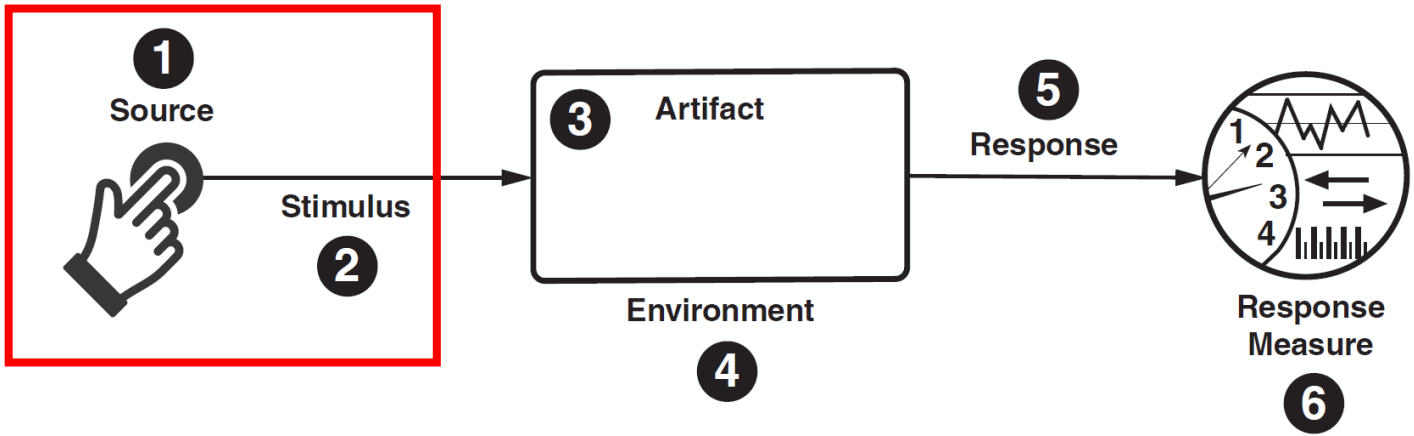
The server should have an uptime of 99.99%

Quality Attribute Scenarios



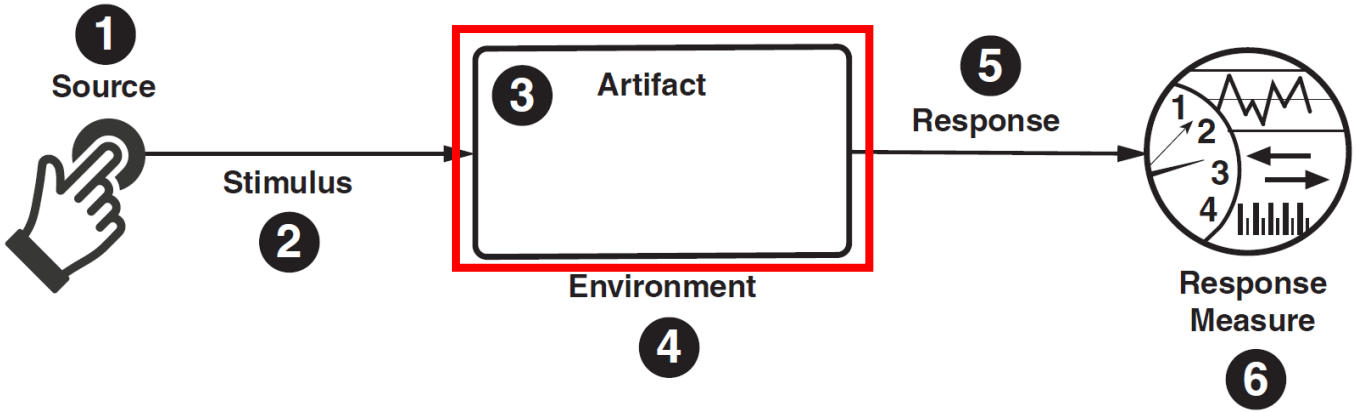
Fundamentals of Software Architecture. An Engineering Approach. Richards and Ford.

Quality Attribute Scenarios



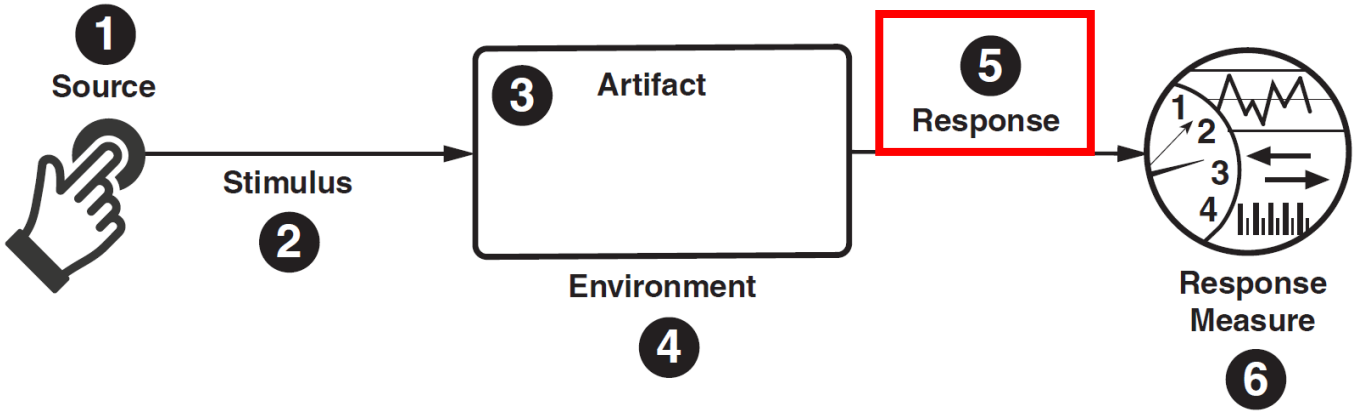
Fundamentals of Software Architecture. An Engineering Approach. Richards and Ford.

Quality Attribute Scenarios



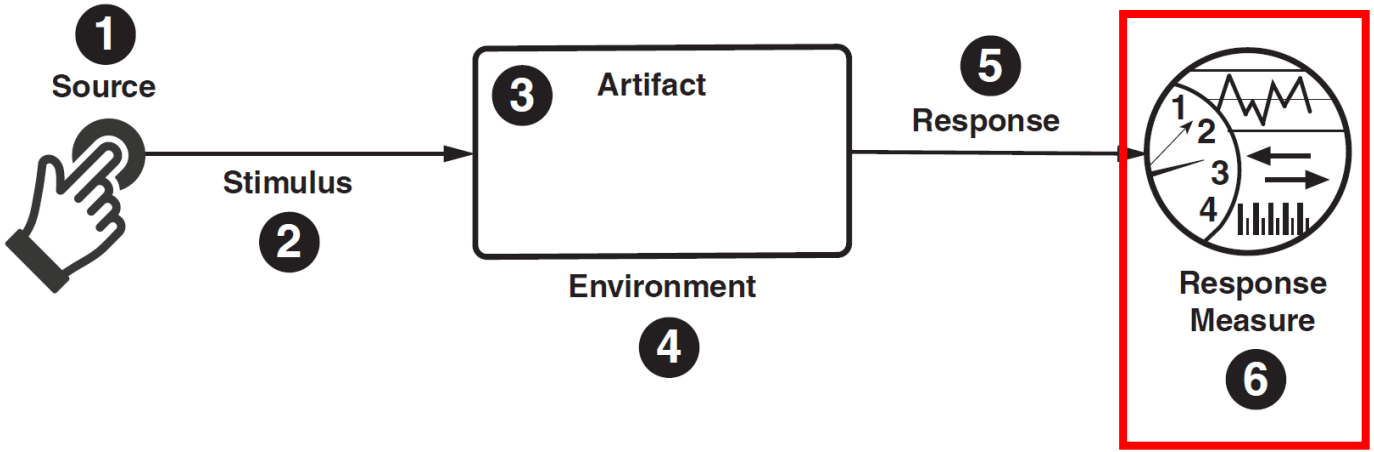
Fundamentals of Software Architecture. An Engineering Approach. Richards and Ford.

Quality Attribute Scenarios



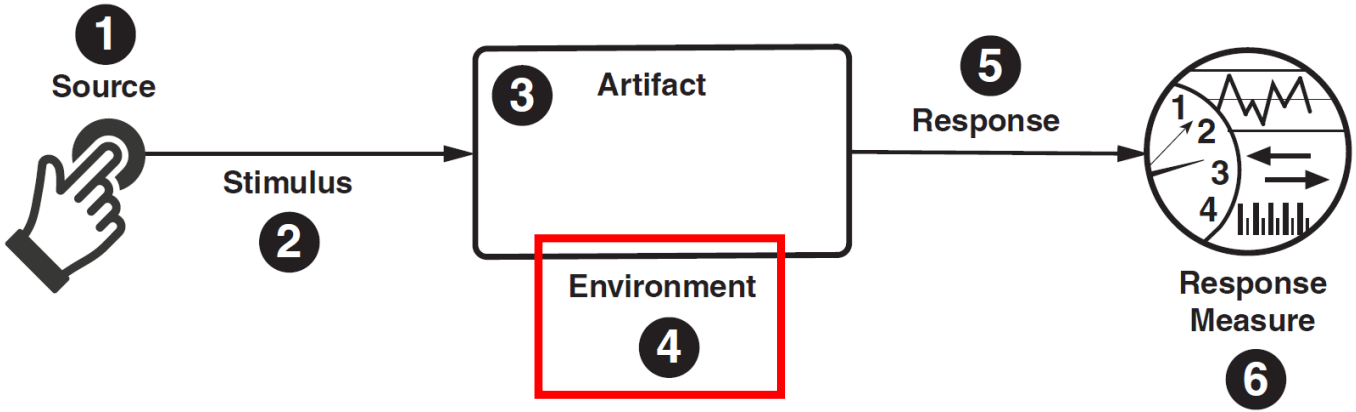
Fundamentals of Software Architecture. An Engineering Approach. Richards and Ford.

Quality Attribute Scenarios



Fundamentals of Software Architecture. An Engineering Approach. Richards and Ford.

Quality Attribute Scenarios

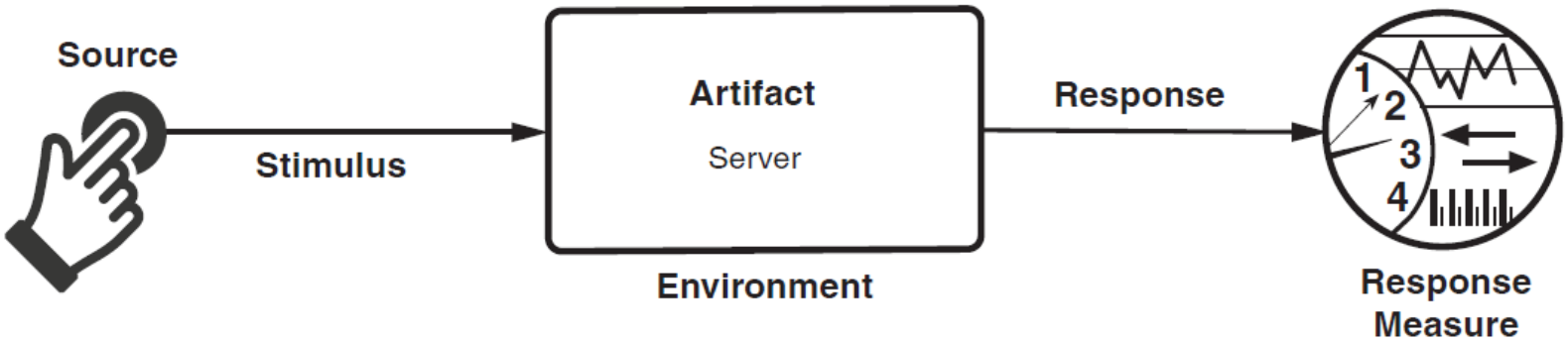


Fundamentals of Software Architecture. An Engineering Approach. Richards and Ford.

Availability 

- **Availability**: system is there and ready to carry out its task
- Related to **security**: a denial-of-service attack is designed to make a system unavailable
- Related to **performance**: it may be difficult to tell when a system has failed or when it is very slow to respond

Quality Attribute Scenarios: Availability



Source	Stimulus	Environment	Response	Response Measure
Server in a server farm	Server fails	Normal operation	System informs operator System continues to operate	No downtime

Availability Metrics: Examples

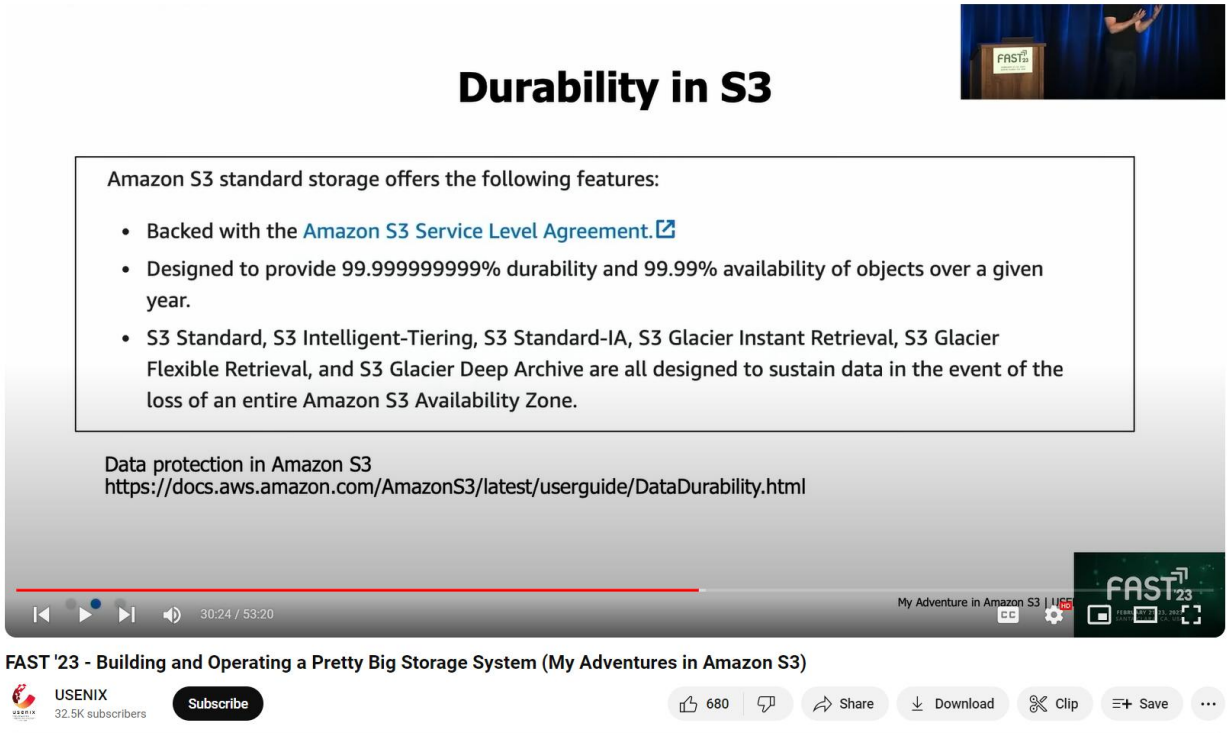
- Availability percentage (e.g., 99.999 percent)
- Time or time interval when the system must be available
- Time to detect the fault
- Time to repair the fault
- Time or time interval in which system can be in degraded mode
- Proportion (e.g., 99 percent) or rate (e.g., up to 100 per second) of a certain class of faults that the system prevents, or handles without failing

Availability

Availability	Downtime/90 Days	Downtime/Year
99.0%	21 hr, 36 min	3 days, 15.6 hr
99.9%	2 hr, 10 min	8 hr, 0 min, 46 sec
99.99%	12 min, 58 sec	52 min, 34 sec
99.999%	1 min, 18 sec	5 min, 15 sec
99.9999%	8 sec	32 sec

Fundamentals of Software Architecture. An Engineering Approach. Richards and Ford.

Example: SLA and Durability in Amazon S3



Service Level Agreements (SLAs)

https://www.youtube.com/watch?v=sc3J4McebHE

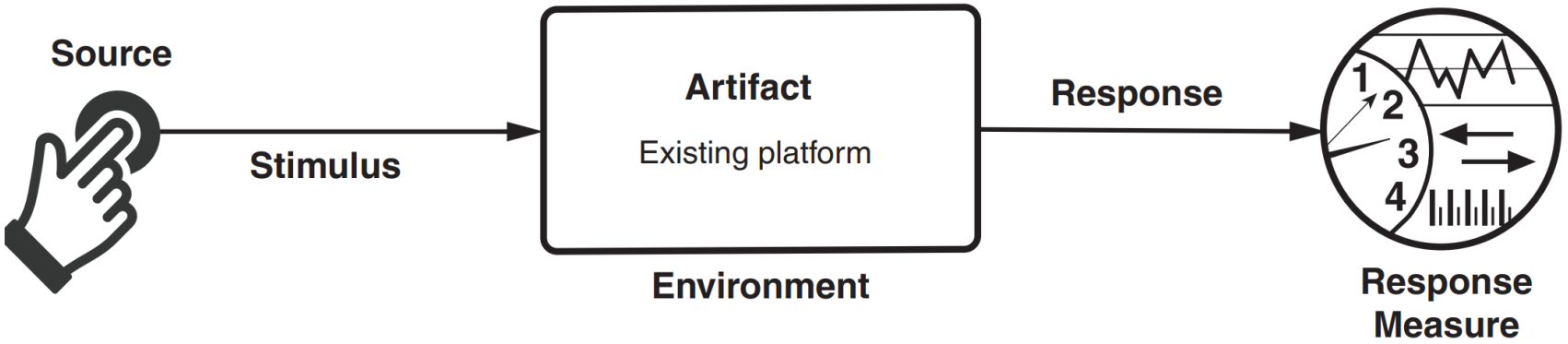
Usability

- **Usability**: how easy it is for the user to accomplish a desired task and the kind of user support that the system provides
- Includes various aspects
  - Learning system features
  - Using a system efficiently
  - Minimizing the impact of user errors
  - Adapting the system to user needs

Usability : Metrics

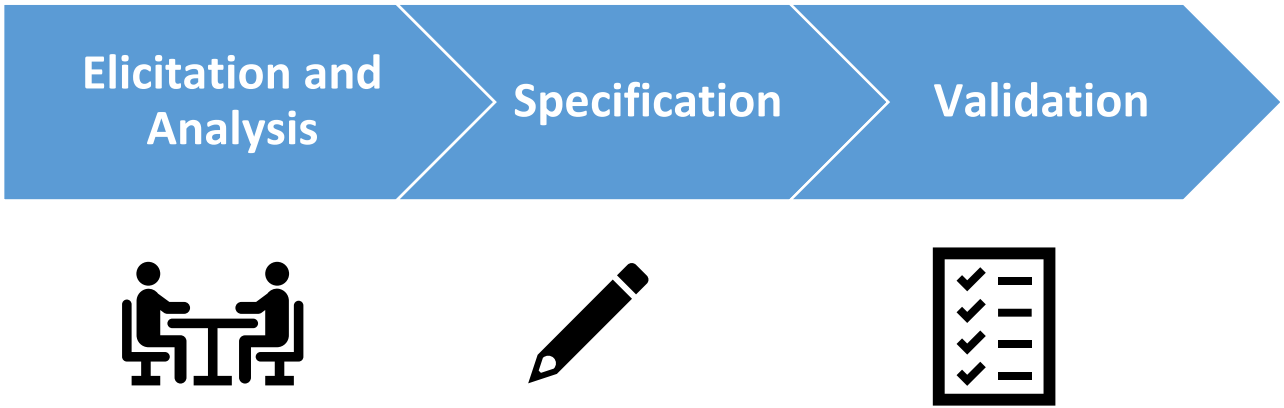
- Task time
- Number of errors
- Learning time
- Ratio of learning time to task time
- Number of tasks accomplished
- User satisfaction
- …

Usability Quality Attribute Scenario

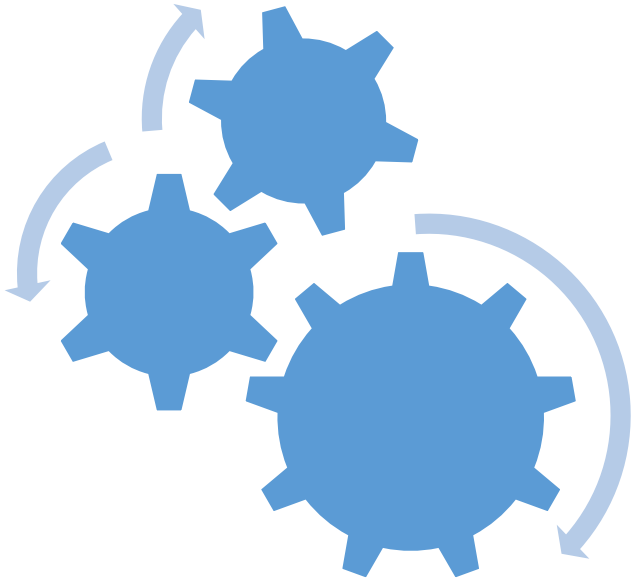


Fundamentals of Software Architecture. An Engineering Approach. Richards and Ford.

Key Activities



Process Models



Process model

Acknowledgements and Main Sources

Summary and Key Points

- Requirements engineering is a “Wicked Problem”
- Snowball sampling to identify stakeholders
- Functional and non-functional requirements
  - Non-functional requirements: quality attributes and constraints
    - Quality Attribute Scenarios