

The Systems Aware Quality Attribute Workshop

Content

- Introduction
 - Systems Thinking vs. Systems Engineering
 - Soft Systems Methodology
 - Quality Attributes and Scenarios
- Overview of SAQAW
 - The steps
 - SAQAW vs. Other methods

What is the Systems Aware Quality Attribute Workshop (SAQAW)?

- „Systems Aware“ – Relies on modeling of the context/organisational System for requirements elicitation
- Uses Systems Thinking (based on SSM) combined with established Software Architecture techniques
- Developed as part of Masters thesis

Software Architecture

- Underlying structure of a system
- Driven by Non-functional requirements expressed as quality attributes
- Most established techniques are scenario-based

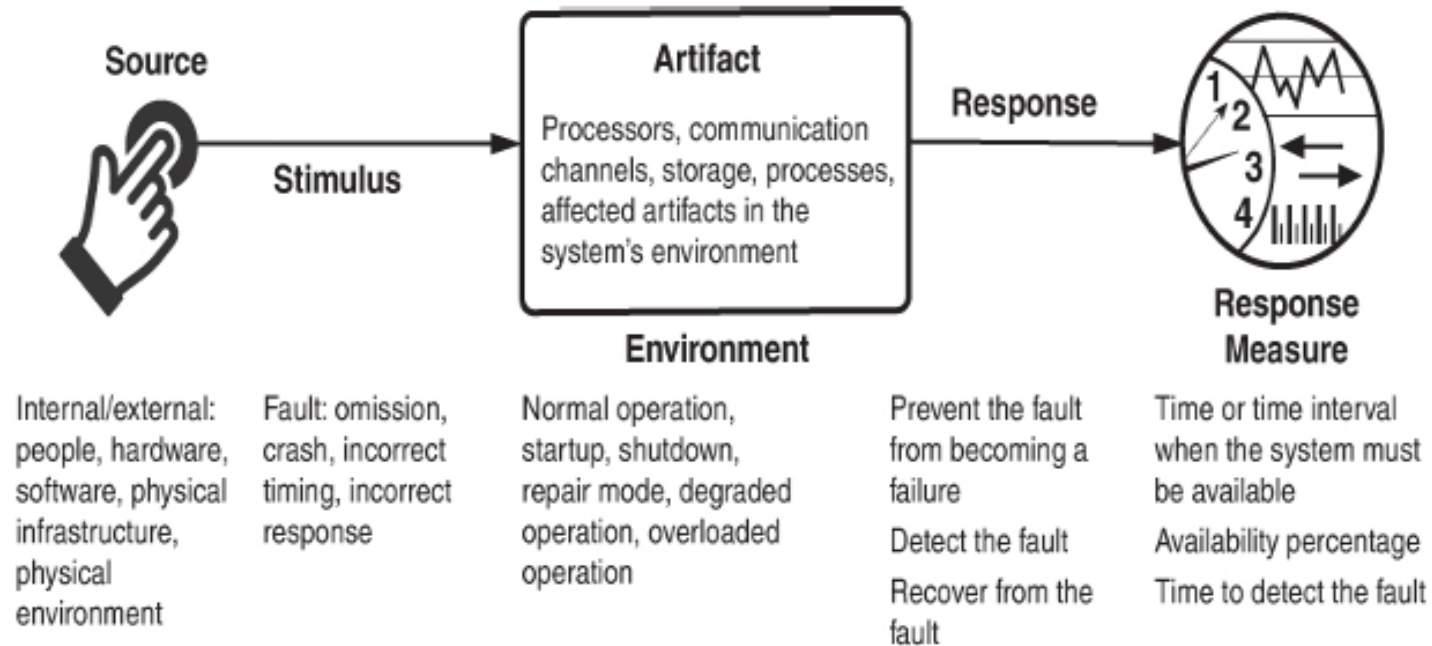
Quality Attributes

- Performance
- Modifiability
- Extensibility
- Reliability
- Deployability
- Testability
-

Characteristic	Subcharacteristics	Short definition
functionality	accuracy	provision of right or agreed results or effects
	compliance	adherence to application related standards or conventions
	interoperability	ability to interact with specified systems
	security	prevention to unauthorised access to data
	suitability	presence and appropriateness of a set of functions for specified tasks
reliability	fault tolerance	ability to keep a given level of performance in case of faults
	maturity	frequency of failure by faults in the software
	recoverability	capability of reestablish level of performance after faults
usability	learnability	users' effort for learning software application
	operability	users' effort for operation and operation control
	understandability	users' effort for recognizing sw. structure and applicability
efficiency	resource behaviour	amount of resources used and the duration of such use
	time behaviour	response and processing times and throughput rates
maintainability	analysability	identification of deficiencies, failure causes, parts to be modified, etc.
	changeability	effort needed for modification, fault removal or environmental change
	stability	risk of unexpected effect of modifications
	testability	effort needed for validating the modified software
portability	adaptability	oportunity for adaptation to different environments
	conformance	adherence to conventions and standards related to portability
	installability	effort needed to install the software in a given environment
	replaceability	opportunity and effort of using software replacing other

Refinement of Quality attributes as shown in
 “Modeling Non-Functional Requirements” Botella et al. (2001)

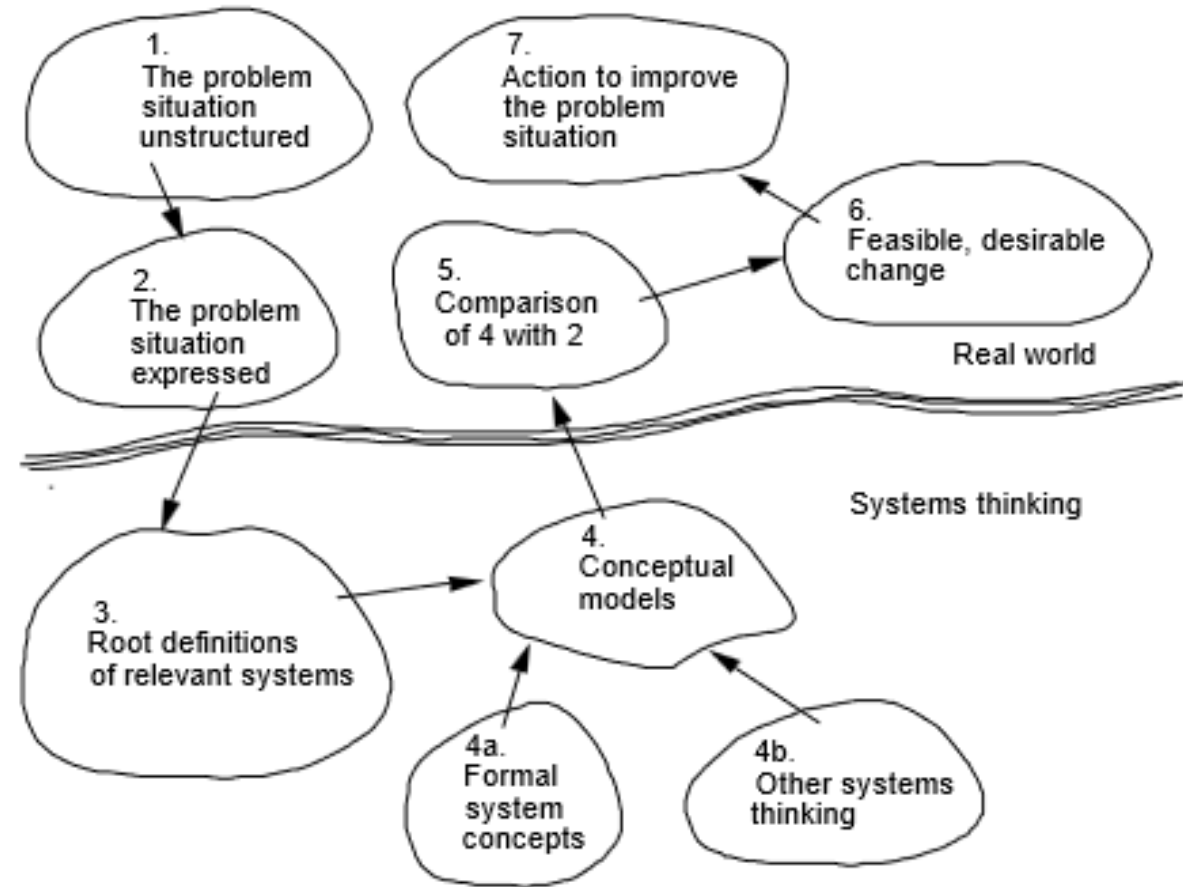
Quality Attribute Scenarios



Structure of Quality Attribute Scenarios taken from Bass et al. (2021)

Soft Systems Methodology (SSM)

- Developed by Peter Checkland in 1981
- Operationalizes Systems Thinking for „ill-defined“ problematic situation
- Rationale: Analysing the real-world as is to inspire purposeful action



7-steps of SSM as defined by Peter Checkland (1981)

Systems Thinking vs. Systems Engineering

ST: A System is a complex situation with socio-technical factors that have certain relations and complexities that need to be analysed as they are to inspire learning and create purposeful solutions

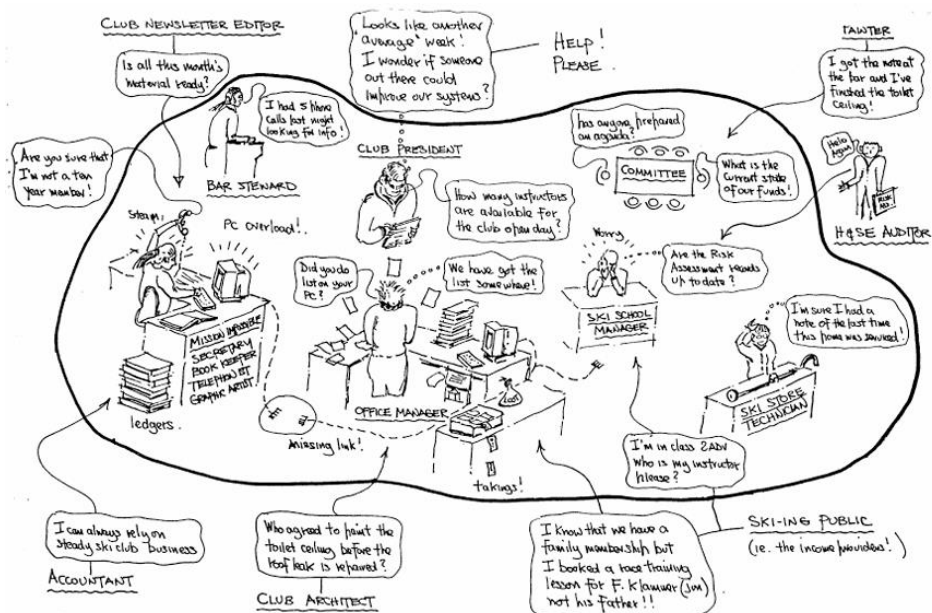
SE: A System needs to be abstracted so a solution can be engineered based on the abstractions, certain aspects of the real-world cannot and should not be expressed

The SAQAW Structure

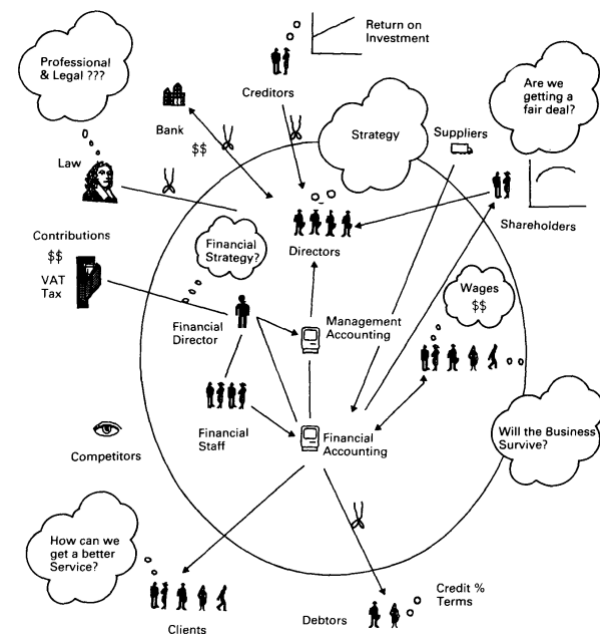
- 8 steps in 2 phases:
- First phase: Soft Systems Phase
- Second phase : Scenario Generation Phase

Soft Systems Phase

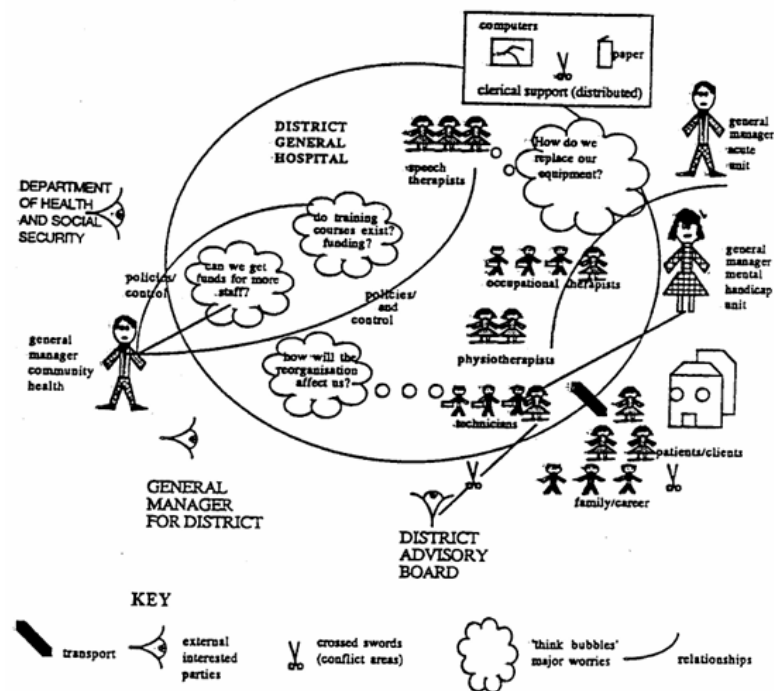
- Step 1: Introductory Presentation
- Step 2: Mission and plan presentation
- Step 3: Rich picture creation
- Step 4: Root Definition



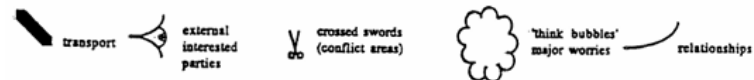
1 2



4 3



KEY



Scenario Generation Phase

- Step 5: Scenario Brainstorming (based on rich picture)
- Step 6: Scenario Consolidation
- Step 7: Scenario Prioritization
- Step 8: Scenario Refinement

References

- Bass, L., Clements, P., & Kazman, R. (2021). Practice, Software Architecture in Practice, Fourth Edition. Boston: Addison-Wesley Professional
- Checkland, P. (1981). Systems Thinking, Systems Practice. John Wiley & Sons Ltd .
- Botella, P., Burgués, X., Franch, X. & Huerta, M. (2001). Modeling Non-Functional Requirements
- Rich Pictures:
 - 1: Bronte-Stewart, M. (1999). Regarding Rich Pictures as Tools for Communication in Information Systems Development. Computing and Information Systems, 83 - 102.
 - 2: Avison, D. E., G. P., & Shah, H. U. (1992). Towards an SSM toolkit: rich picture diagramming. European Journal of Information Systems, 397 - 408.
 - 3: Avison, D., & Fitzgerald, G. (2002). Information Systems Development: Methodologies, Techniques and Tools. McGraw Hill Higher Education.
 - 4: Patching, D. (1990). Practical Soft Systems Analysis. Financial Times.