IIT KANPUR
Indian Institute of Technology Kanpur

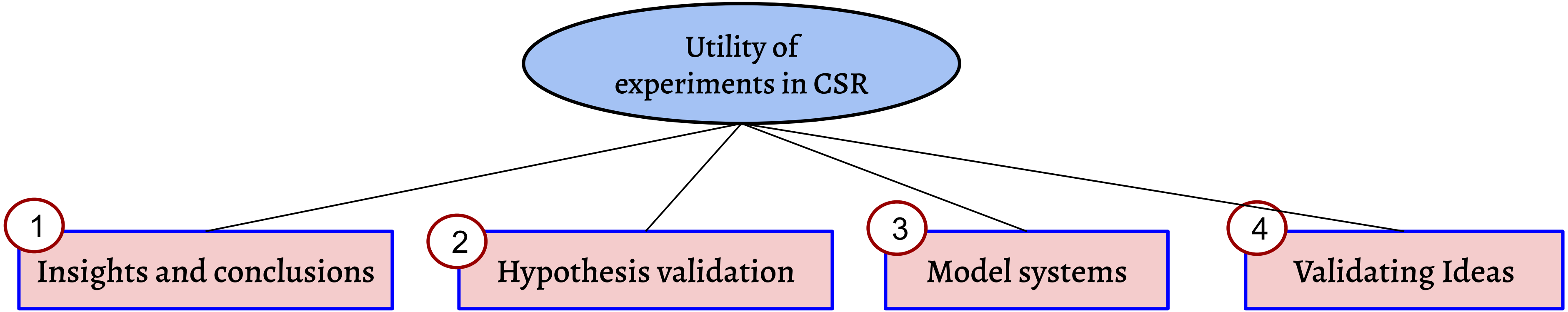
CS888: Introduction to Profession and Communication

System Research: Experiments

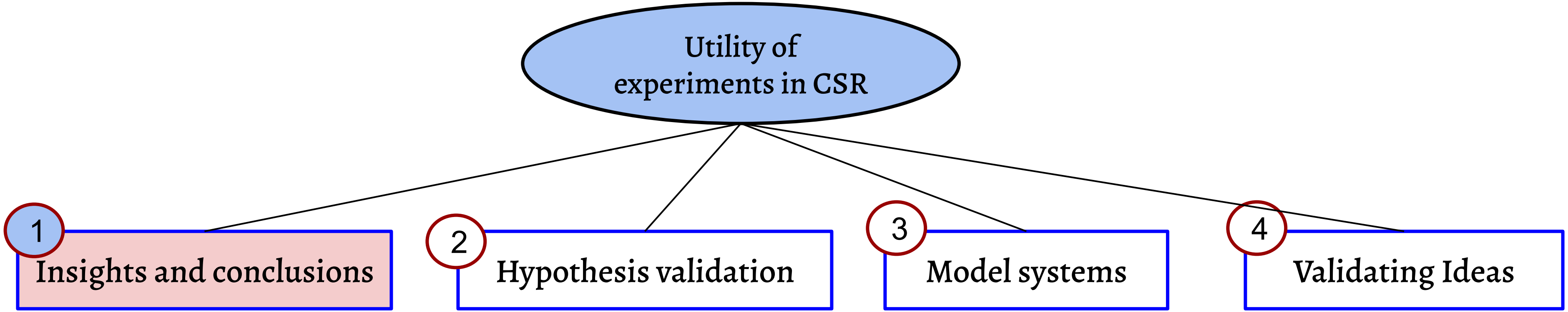
Debadatta Mishra, CSE, IIT Kanpur

Experiments in computer system research

A broad grouping of different ways experiments are used for CS system research

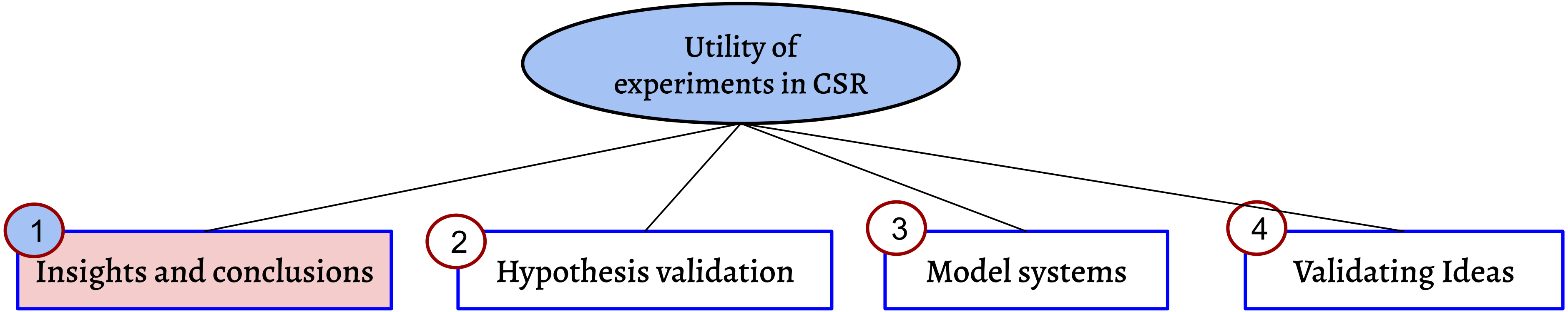


Experiments in computer system research

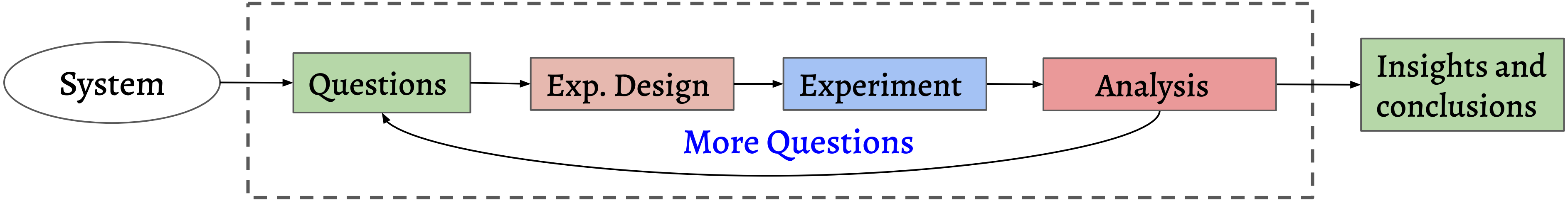


- Quantification of different aspects such as performance, resource cost, and their tradeoffs
- Generalized observations for different usage (workload) scenarios
- Facilitates generating novel ideas for improved system design

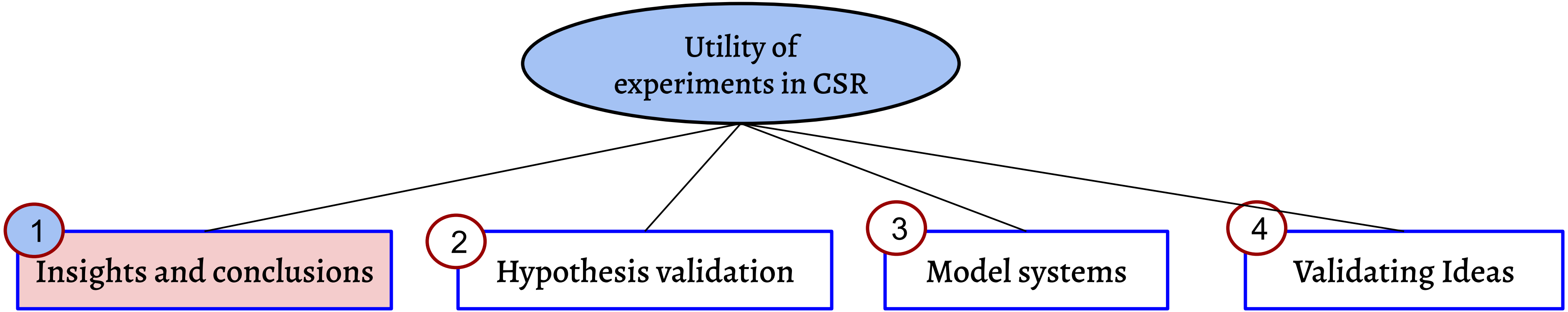
Experiments in computer system research



- Quantification of different aspects such as performance, resource cost, and their tradeoffs
- Generalized observations for different usage (workload) scenarios
- Facilitates generating novel ideas for improved system design



Experiments in computer system research

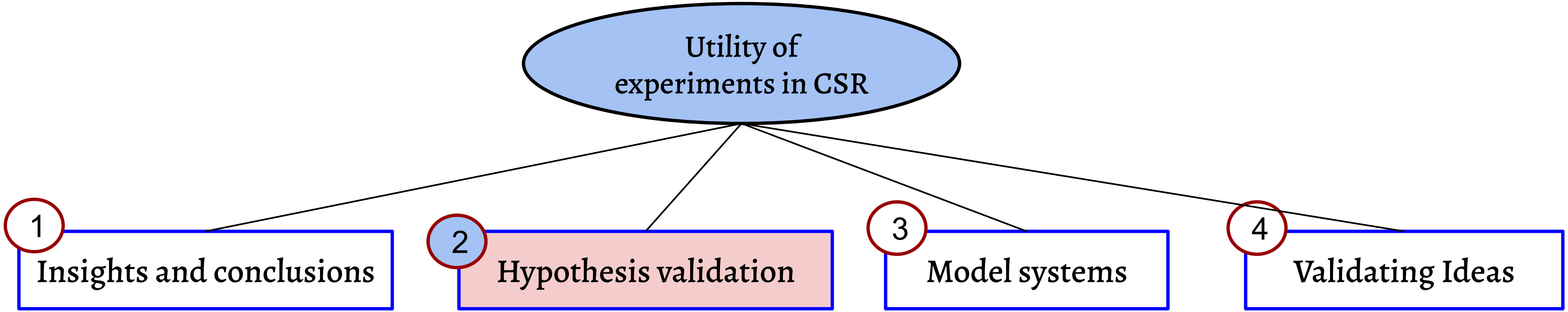


Typical usage

- Empirical analysis
- Root-cause analysis
- Deciding design parameters and tradeoffs

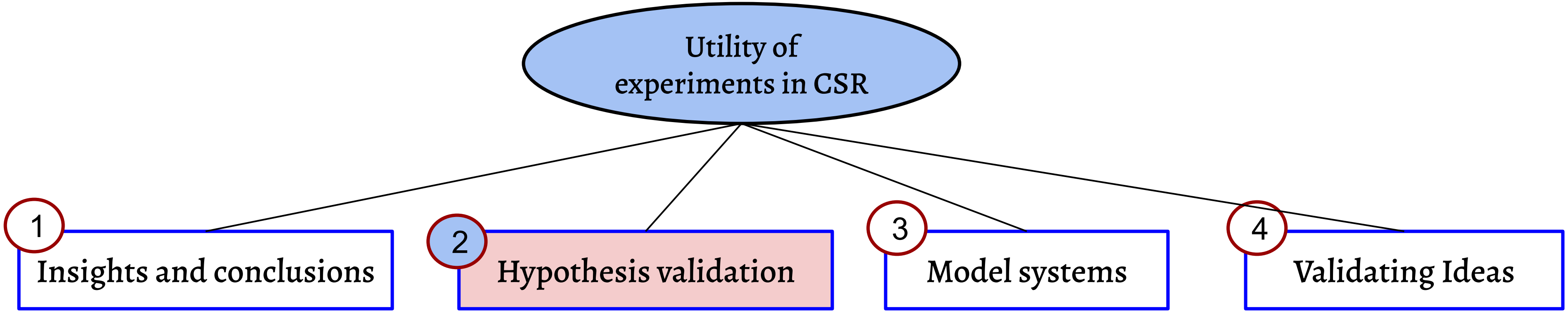
Example: vee2021

Experiments in computer system research

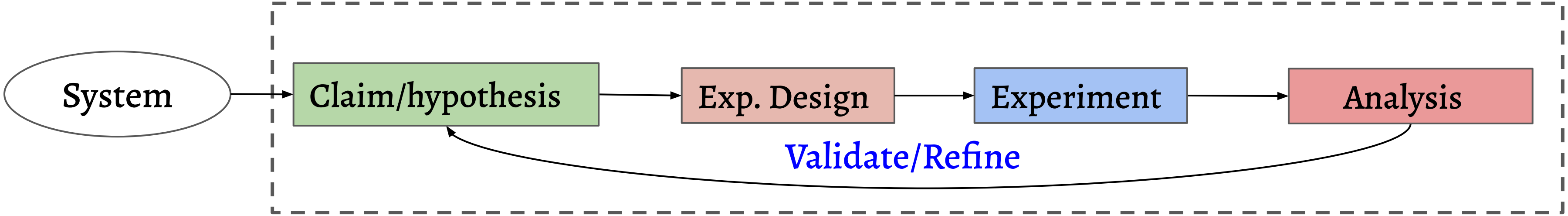


- To establish intuitive/qualitative claims by providing quantitative support
- Can be targeted i.e., generalization requirements can be relaxed to a certain extent

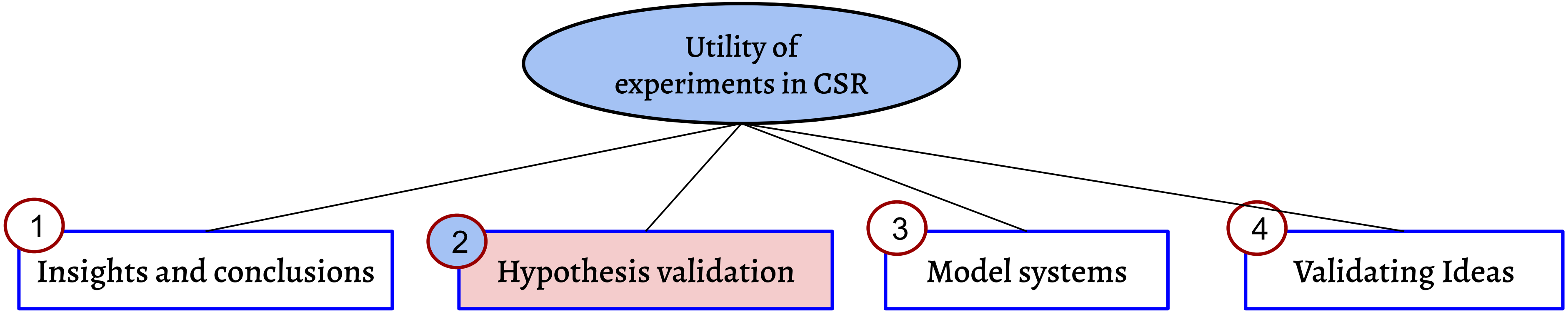
Experiments in computer system research



- To establish intuitive/qualitative claims by providing quantitative support
- Can be targeted i.e., generalization requirements can be relaxed to a certain extent



Experiments in computer system research

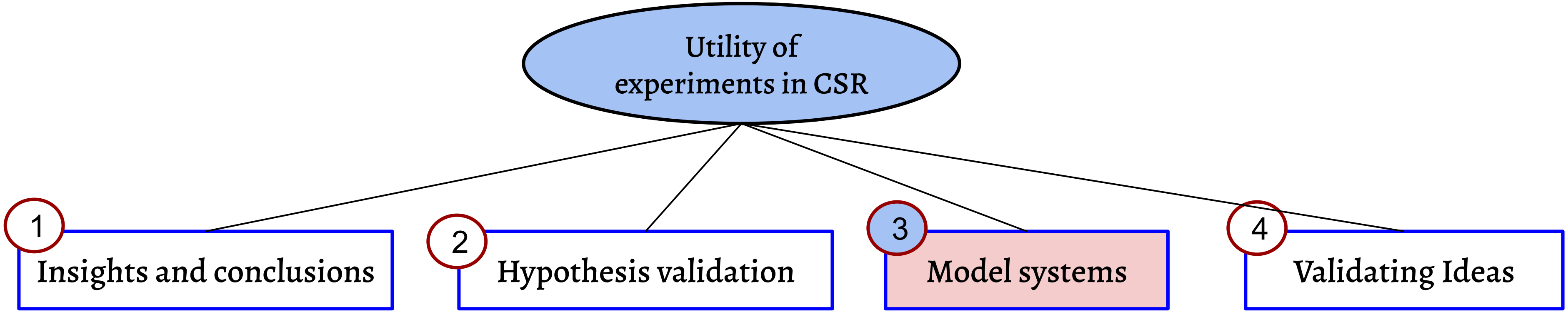


Typical usage

- Motivating/establishing problems
- Establishing design choice rationale

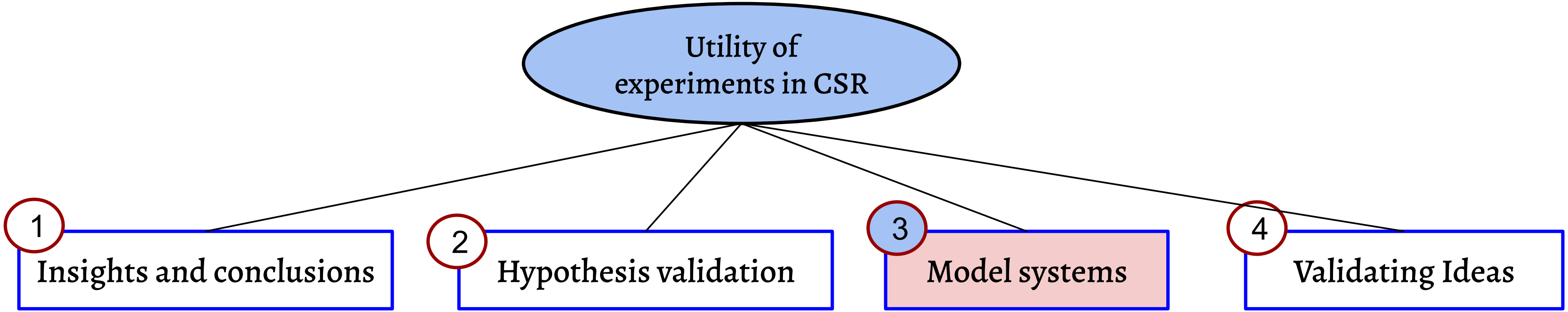
Example: Catalyst

Experiments in computer system research

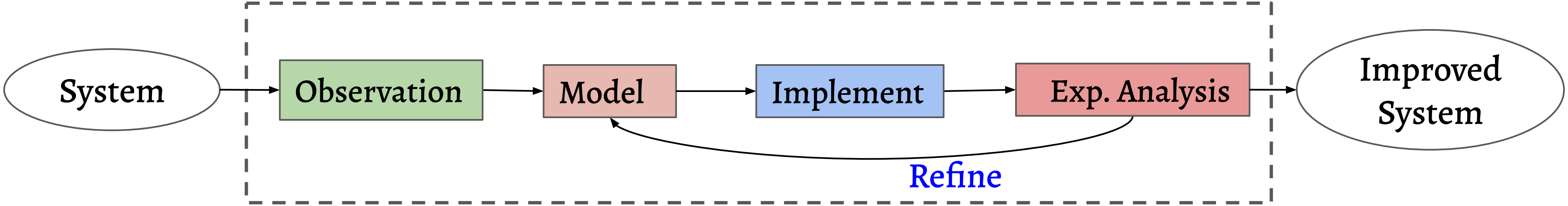


- Model system OR performance and/or cost of a system using experimental analysis
- Typically experimental modelling is the next step after performing Step 1

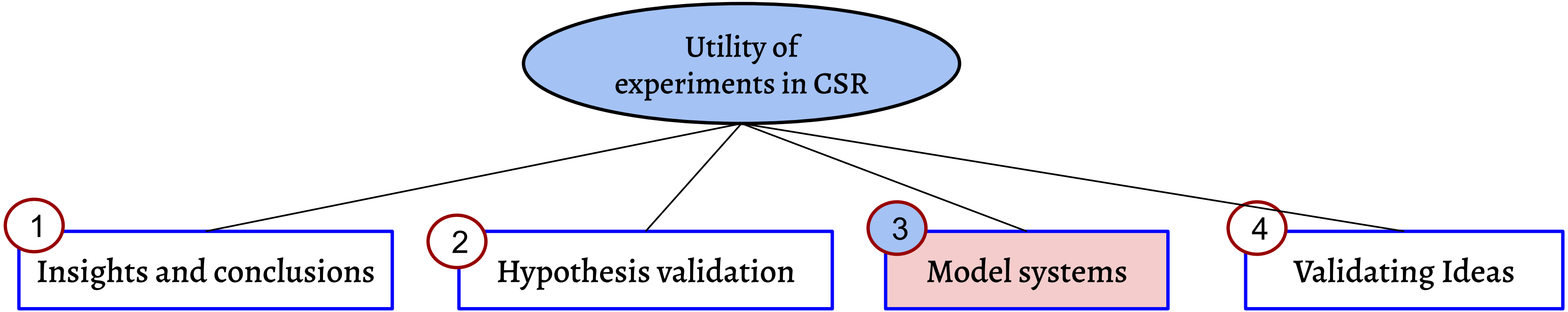
Experiments in computer system research



- Model system OR performance and/or cost of a system using experimental analysis
- Typically experimental modelling is the next step after performing Step 1



Experiments in computer system research

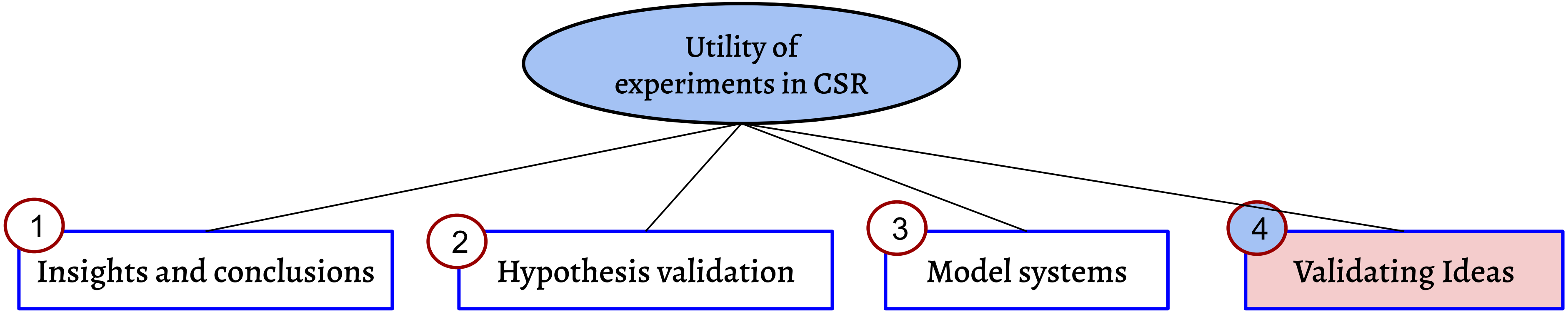


Typical usage

- Policy research
- Performance prediction

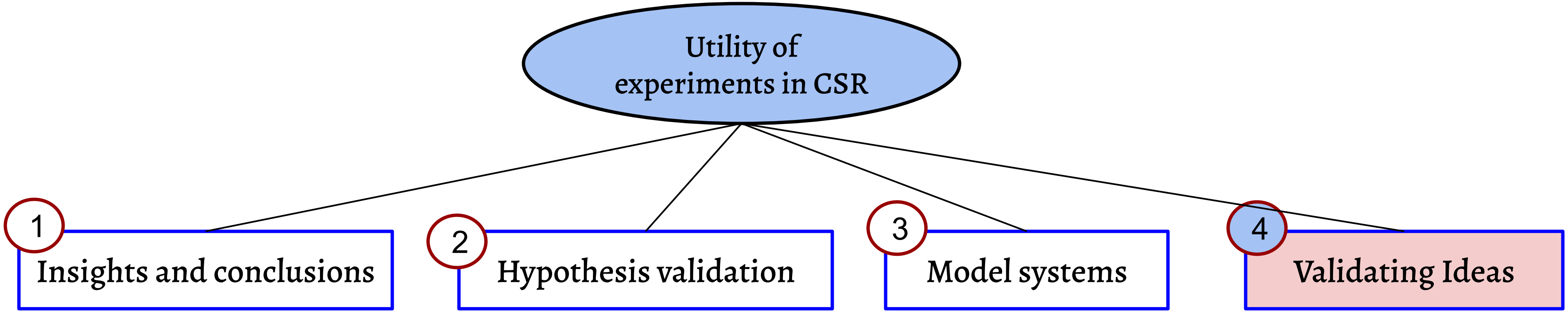
Example: MigModel

Experiments in computer system research

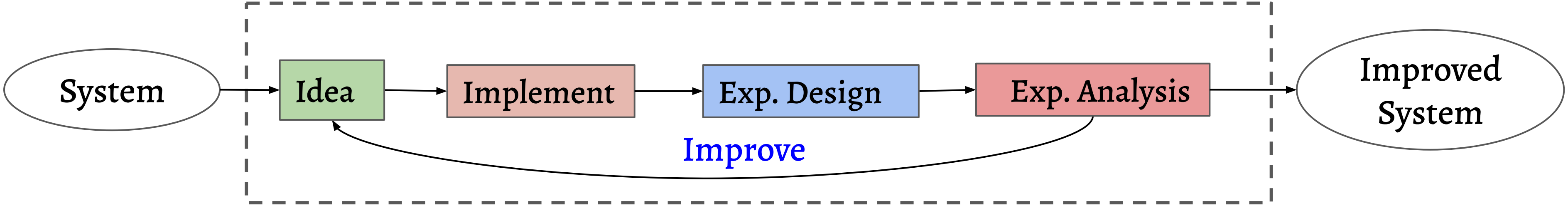


- Typically new ideas are generated by observations and insights
- The first iteration most likely does not work out!

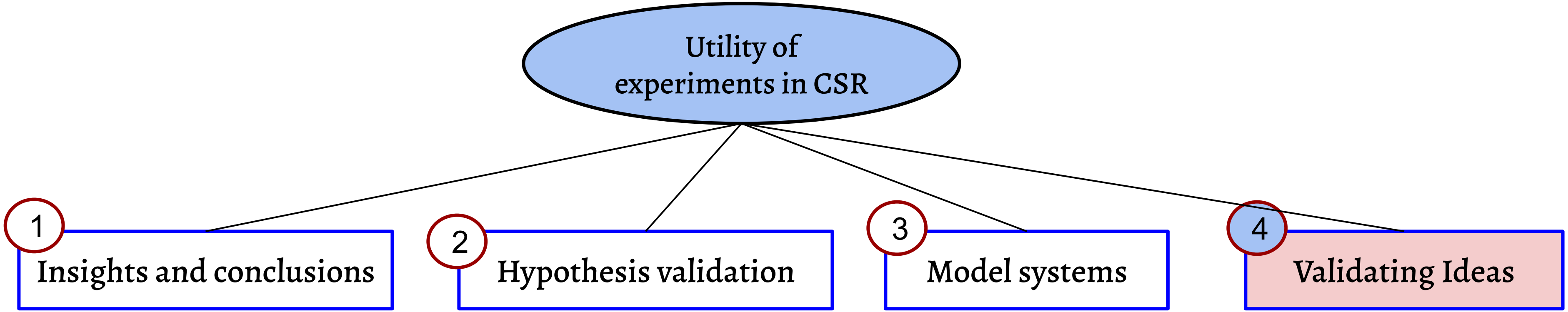
Experiments in computer system research



- Typically new ideas are generated by observations and insights
- The first iteration most likely does not work out!



Experiments in computer system research

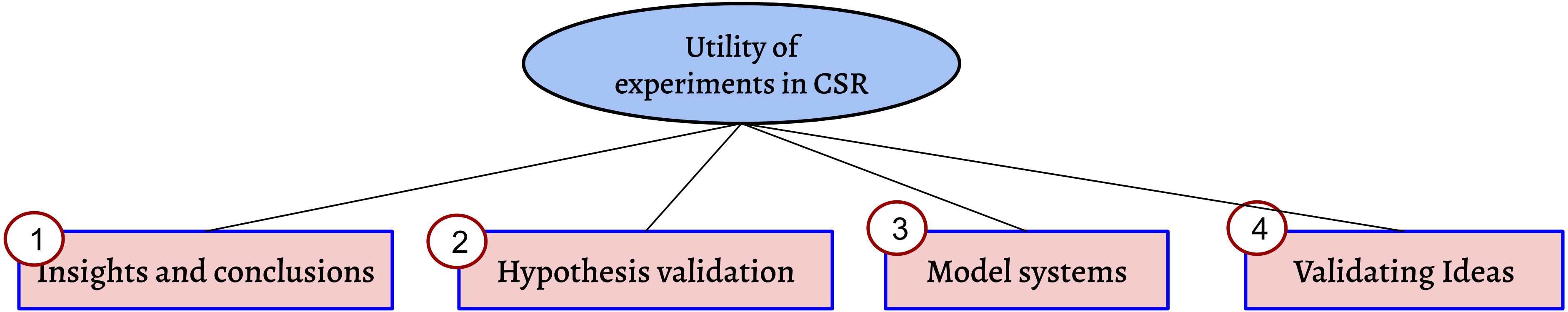


Typical usage

- Evaluate proposed systems/ideas
- Comparative analysis

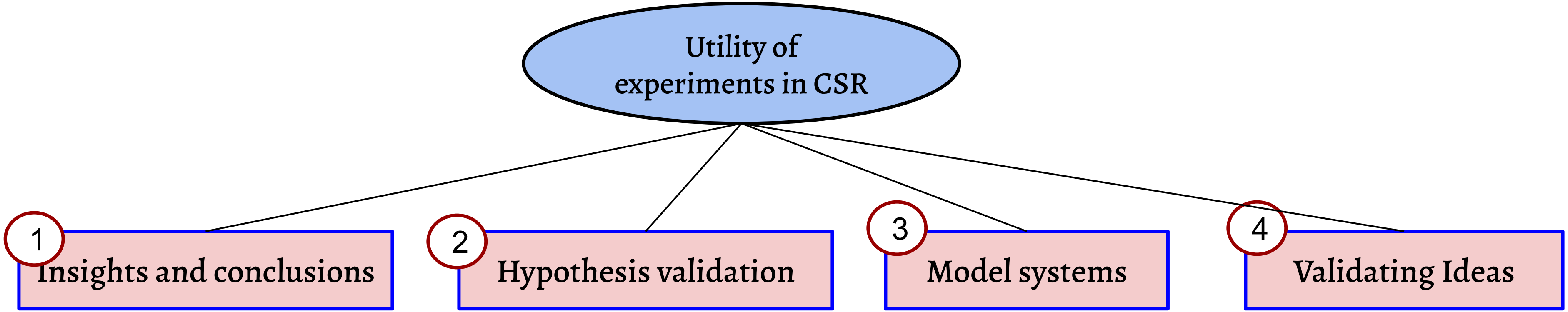
Example: vee2021

Experiments in computer system research



Challenges of using experimental methods in CSR?

Experiments in computer system research

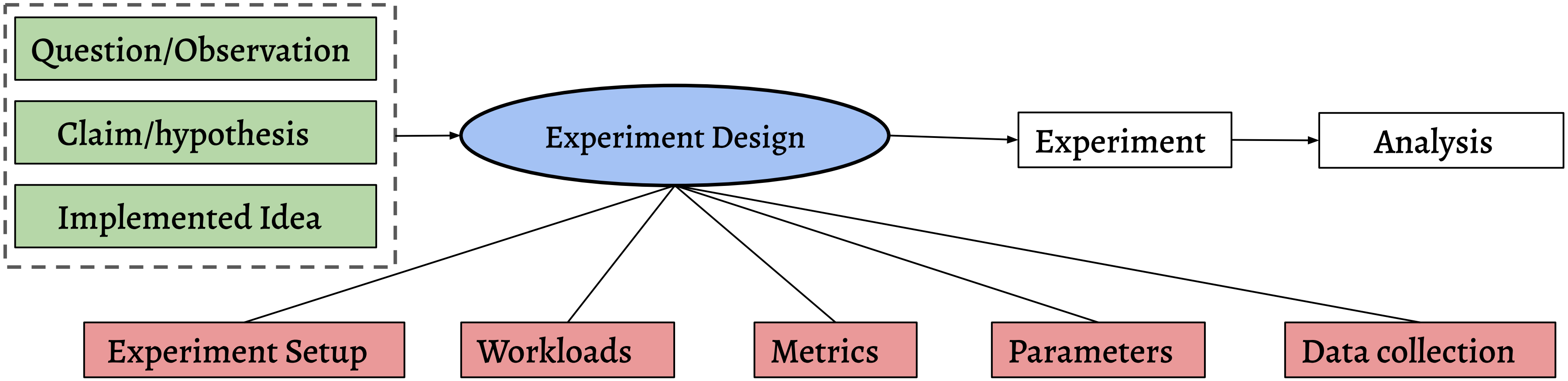


Challenges of using experimental methods in CSR?

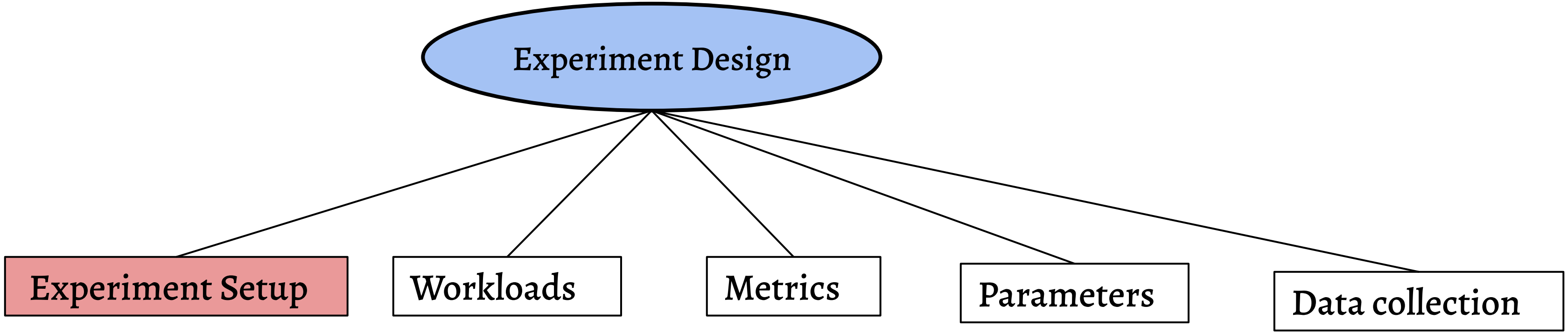
- Correctness: Many complex interactions, need clarity in designing experiments
- Reproducibility: The ecosystem is volatile, comprehensive documentation helps
- Generalizing findings: System dependent, comparative conclusions are more suitable
- Objectivity: Human bias can give misleading conclusions, use best practices

Designing experiment: process and best practices

Designing good experiments is one of the crucial aspects in CSR

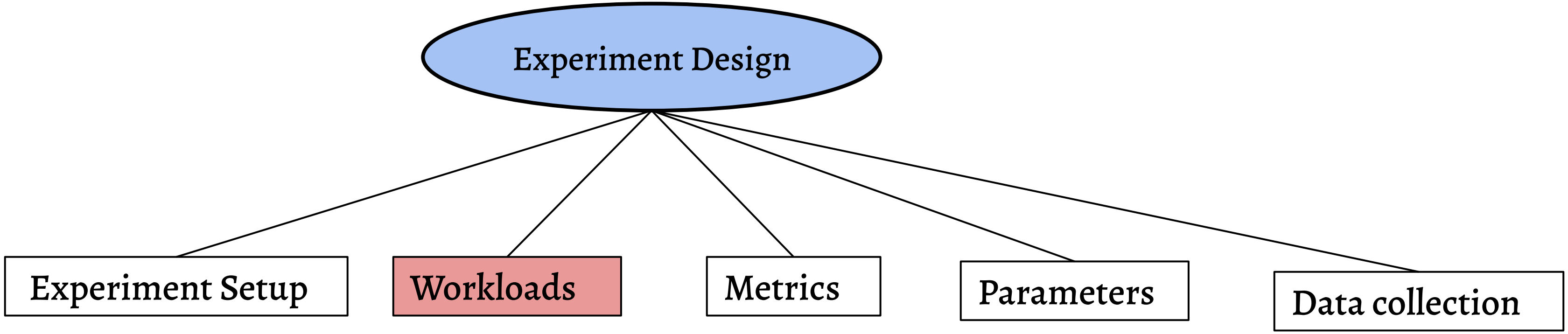


Designing experiment: Experiment setup



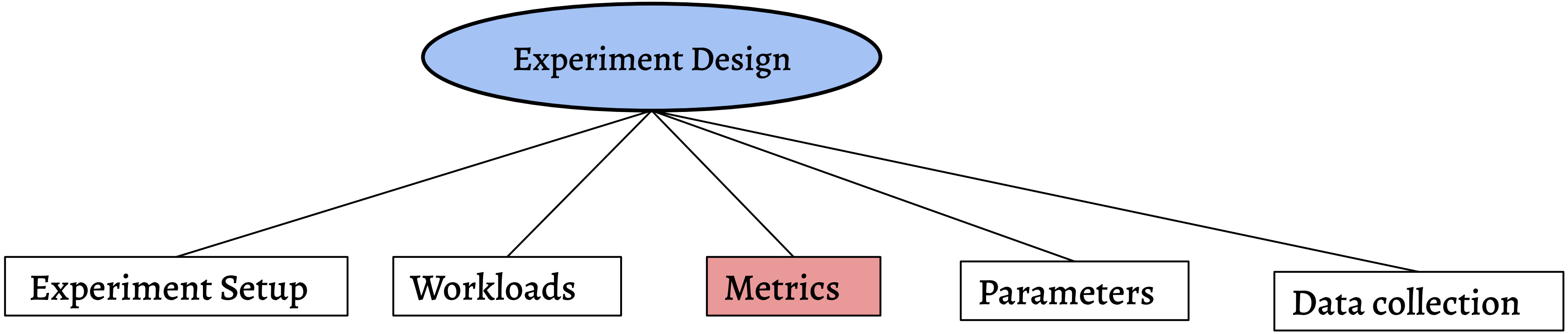
- Create an experiment setup representative of the system and suitable to meet the desired objective of the experiment
- Best practices
  - Keep it close to the real system and define system under test (SUT) precisely
  - Try to reduce the noise as much as possible
  - Note down all sw/hw config, If any system config is not default, note down
  - Clearly define the states (e.g., start of experiment)

Designing experiment: Workloads



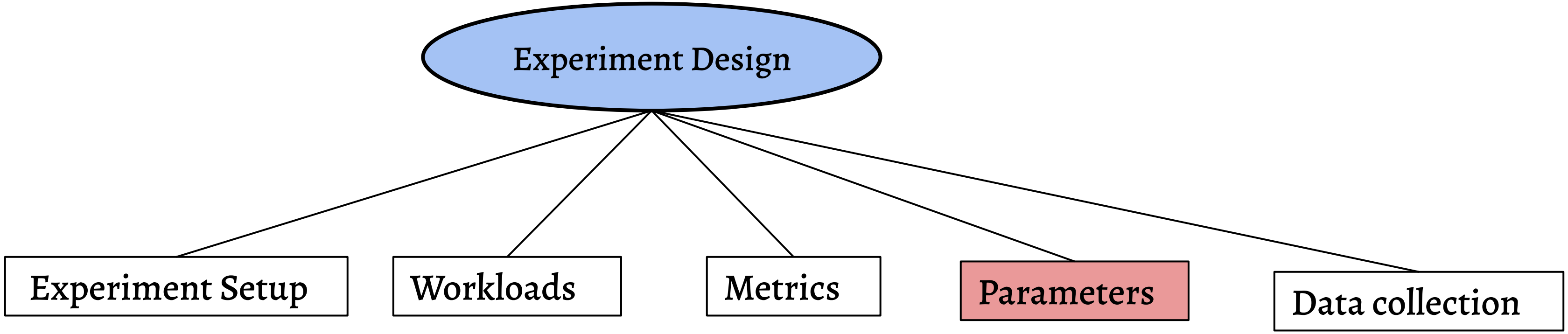
- Select a set of workloads, not only to study the system behaviour in question but also other seemingly dependant properties/resources
- Best practices
  - Realistic benchmarks, note the benchmark parameters if they are not default (also justify why default not used)
  - Should cover all cases e.g., favorable, moderate and extreme workloads
  - Microbenchmarks can be used to study specific aspect in an isolated manner

Designing experiment: Metrics



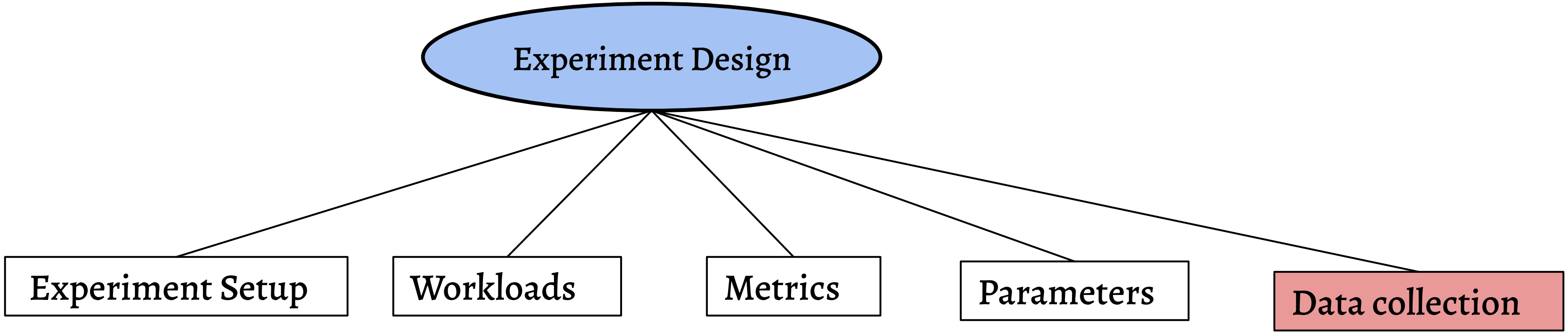
- Identify the metrics of interest based on the question/hypothesis/idea
- Best practices
  - Ideally, metrics should answer the question, establish/nullify the hypothesis
  - Easy to measure proxies should be avoided
  - Should include: what else can be impacted?
  - Recall that surprises can be good! Expected result may not be very interesting.

Designing experiment: Parameters



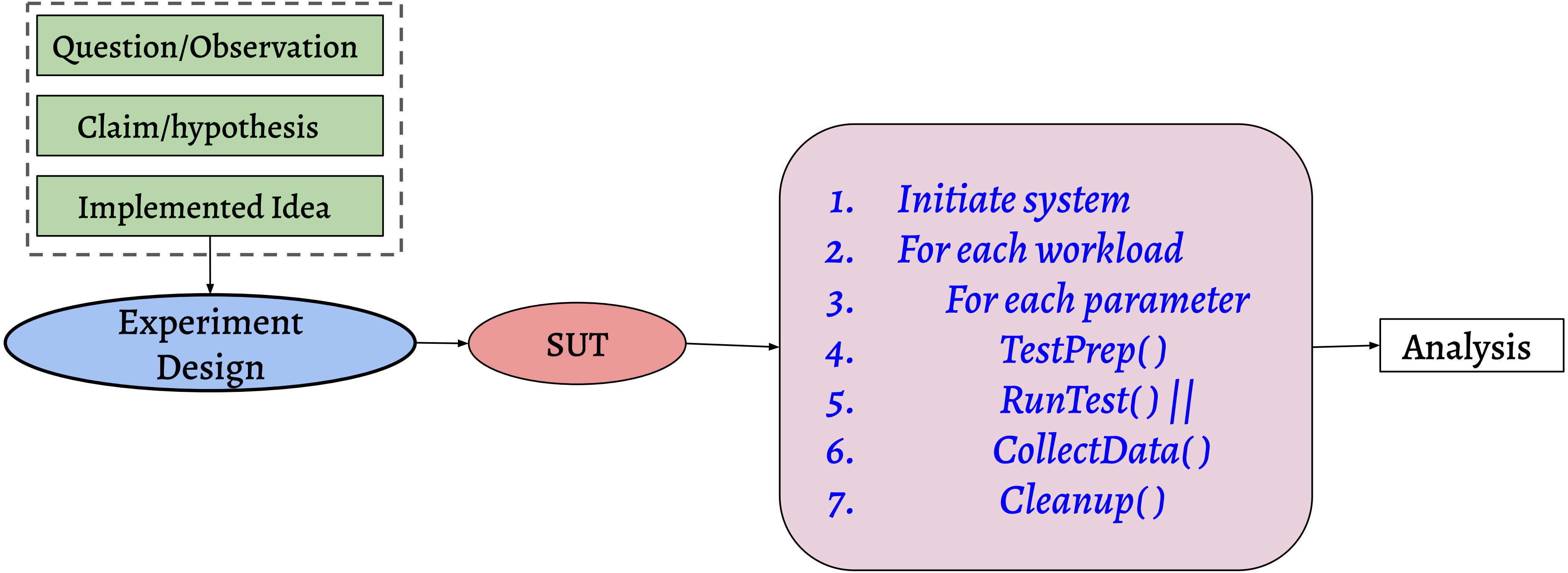
- Create a list of parameters that can influence the outcome of the experiments
- Best practices
  - If not sure, include the parameter in the list
  - Do not knowingly ignore parameters (avoid human bias!)
  - Explosion of parameter space may become intractable, group related params
  - Varying too many parameters in an experiment may not work, study in isolation

Designing experiment: Data collection



- Identify all information to be collected during the experiment, create automation
- Best practices
  - Data collection should have minimal interference with the experiment
  - Raw data is valuable, do not discard or summarize it during the experiment run
  - Always try to collect as much data as possible, avoids experiment repetition

Designing experiment: Putting everything together



- Reproducibility: Should be able to access the experiment setup details, configurations, customizations, parameter, raw data along with the process of summarization

Analysis

- If you formulated the hypothesis/question precisely, analysis can be targeted; too much information may make you feel lost
- Summarization using statistical techniques
  - Mean (AM vs GM), variance
  - Distribution: scatter plots, CDF, tail of distributions
- Plotting and tabulation
  - Plots can reveal many subtle behaviors; plotting tools (e.g., gnuplot, matplot)

- Do not discard anomalies as outliers ⇒ may lead to new insights
- Keep an open eye for inconsistencies ⇒ may reveal a buggy design, implementation or a error prone experiment setup

Reporting experiments

- What we can not see, others may! $\longrightarrow$ Report comprehensively
- Detailed explanation of setup, parameters, workloads and metrics
- More is better for “reproducibility”
- “Reproducible experiments” $\longrightarrow$ closer to truth
- Keep your setups ready for reuse (e.g., artifact validation)

- Conclusions should be based on factual findings
- If you can not explain something, need more experiments/analysis!

Sins![#]

- Small experiment duration → reporting non-steady state results
- Not explicitly exposing the "weak links"
- Omitting workload results because they do not suit the "narrative"
- Selective reporting or data hiding
    - Showing the performance when everything is a cache hit
- Downplaying overheads
    - CPU usage increase from 7% to 15% is not 8% increase!
- Average without commenting on the variance
- Using wrong baselines

Systems Benchmarking Crimes: https://www.cse.unsw.edu.au/~gernot/benchmarking-crimes.html

Case study: A full cycle of experimental analysis

- KB-CCGrid2023 Problem Scope: Multi-user application cloud
- Observations and Questions: Can we use FaaS? Cost and performance
- Motivation: Resource (memory) overheads, tradeoff in performance
- Design and implementation: Thorough analysis helps in preparing a strategy for designing a solutions
- Experimental evaluation
  - Evaluation questions, setup details
  - Comparative evaluation, causal analysis, analysing design elements in isolation, scalability in a real setup