Empirical Investigation

CHAPTER 4

Software Measurement Validation

- ✓ Software **verification** is the process of evaluating and checking whether the software being developed or modified correctly according to the specified requirements. Verification includes all the activities associated with producing high quality software, i.e.: testing, inspection, design analysis, specification analysis, and so on. In other words, it answers the question, "**Are we building the software right**?"
- ✓ Software **validation**, on the other hand, is the process of evaluating a software system during or at the end of the development process to determine *whether it satisfies the intended user requirements*. In other words, it answers the question, "**Are we building the right software**?"

Software Measurement Validation

A large number of software measures in the literature aims to capture information about a wide range of attributes.

Same attribute (e.g., cost, size, or complexity) can be measured in very different ways. So, finding the best measure/metric is difficult.

We must ensure that the measures/metric we use actually capture the attribute information we seek. Thus validation of measures/metric is important.

Two types of measuring:

- Measure or measurement systems are used to assess an **existing entity by numerically characterizing** one or more of its attributes.
- **Prediction systems** are used to **predict** some attribute of a future entity, involving mathematical model.

We can say that a **measure** is "*valid*" if it accurately characterizes the attribute it claims to measure

A **prediction system** is "*valid*" if it makes accurate predictions.

Validating Prediction Systems

- **Definition**: *Validating a prediction system* in a given environment is the process of establishing the accuracy of the prediction system by empirical means; that is, by comparing model performance with known data in the given environment.
- Thus, validation of prediction systems involves **experimentation** and **hypothesis testing**
- The degree of accuracy acceptable for validation depends on several things
  - Person doing assessment
  - Deterministic prediction system
  - Stochastic prediction systems

  - Prediction systems for software cost estimation, effort estimation, schedule estimation, and reliability estimation are very stochastic, as their margin of error are very large

  Some model use within 20% accuracy as acceptable range, Some project manages may find this range to be too large to be useful planning.

Validating Measures/metrics

- **Definition:** *Validating a software measure* is the process of ensuring that the measure is a proper numerical characterization of the claimed attribute by showing that the representation condition is satisfied.
- This type of validation is central to the representational theory of measurement.
- That is, we want to be sure that the measures we use reflect the behavior of entities in the real world.
- If we cannot validate the measures, then we cannot be sure that the decisions we make based on those measures will have the effects we expect.

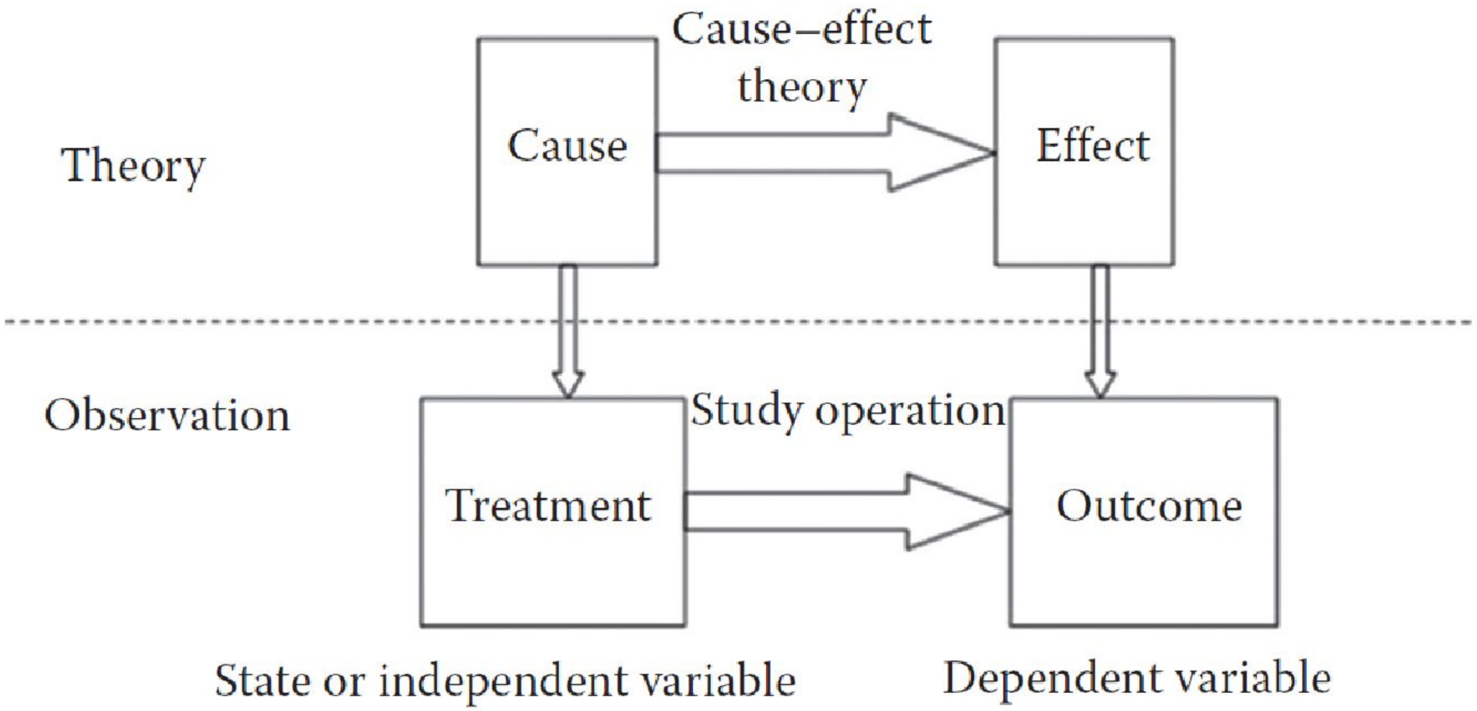
We must choose a valid measurement/metrics which follow the three basic mathematical properties (we discussed it earlier)

Empirical Studies

- Selecting the best or optimal tool or technique requires some empirical support
- **Empirical research** is based on observed and measured phenomena and derives knowledge from actual experience rather than from theory or belief.
- Empirical studies are conducted to test the theory or belief
- Empirical studies do not prove that a theory is true, rather they provide further evidence to support or refute the theory
- An empirical study examines some specific sample or observation of **all of the possible values** of the variable involved **in a cause-effect relationship**.

Empirical Studies

*** A cause-effect relationship is a relationship in which one event (the cause) makes another event happen (the effect)*



Empirical Studies

Key characteristics involved in designing empirical studies include:

1. The level of control of **study variables** (e.g., population, behavior, or phenomena**)** that determines the **appropriate type of study**
2. Study goals and hypotheses
3. Maintaining the control of variables
4. Treads to validity
5. The use of human subjects

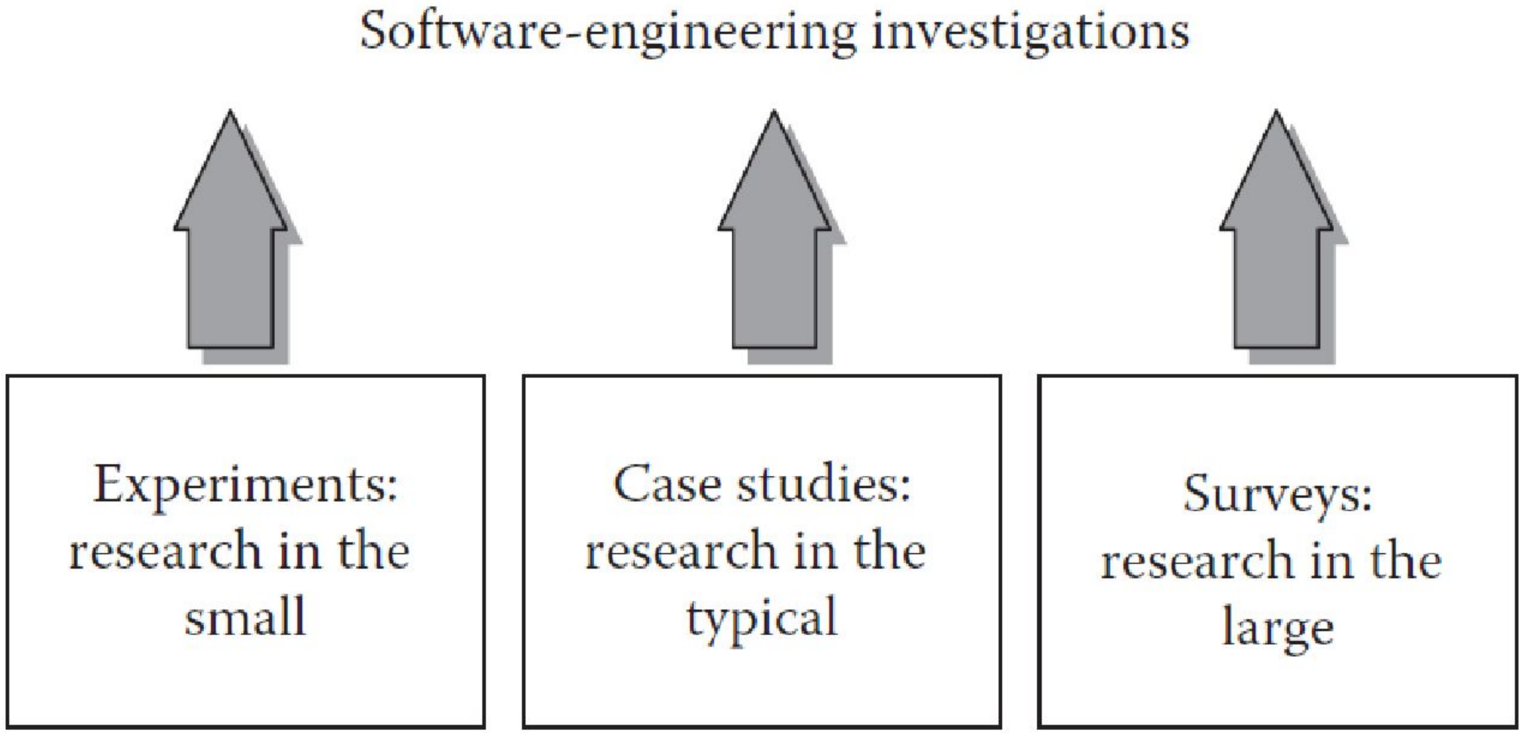
1. Control of Variables and Study Type

Controlled experiments involve the testing of well-defined hypotheses concerning the postulated effects of independent variables on dependent variables

Sometimes the input variables can not be controlled, e.g., in the research area of astrophysics and geology.

Ethical and legal issues are also factors that prevent conducting experiments.

Investigation Type



Software-engineering investigations

Experiments: research in the small

Case studies: research in the typical

Surveys: research in the large

Experiments: When research is small

● Experiments require a great deal of control, they tend to involve small numbers of people or events

● Controlled experiments tend to be conducted in academia or research labs

● Rarely performed inside software industrial environment as controlling for

confounding factors is hard

Case Studies: Research is Typical

- Empirical studies that involve ***observations*** where potential confounding variables cannot be controlled and/or subjects cannot be assigned to treatment or control groups are called ***observational studies, natural experiments, and/or quasi-experiments***
- A case study documents activity by identifying key factors (inputs, constraints, and resources) that may affect the outcomes of a research
- Empirical studies in software engineering are case studies that involve the use of a tool or technique on projects without random assignment of subjects to projects and control of all other variables
- Experiments are controlled, case studies are observational data collection and study

Experiments vs Case Studies (Table is read across and up)

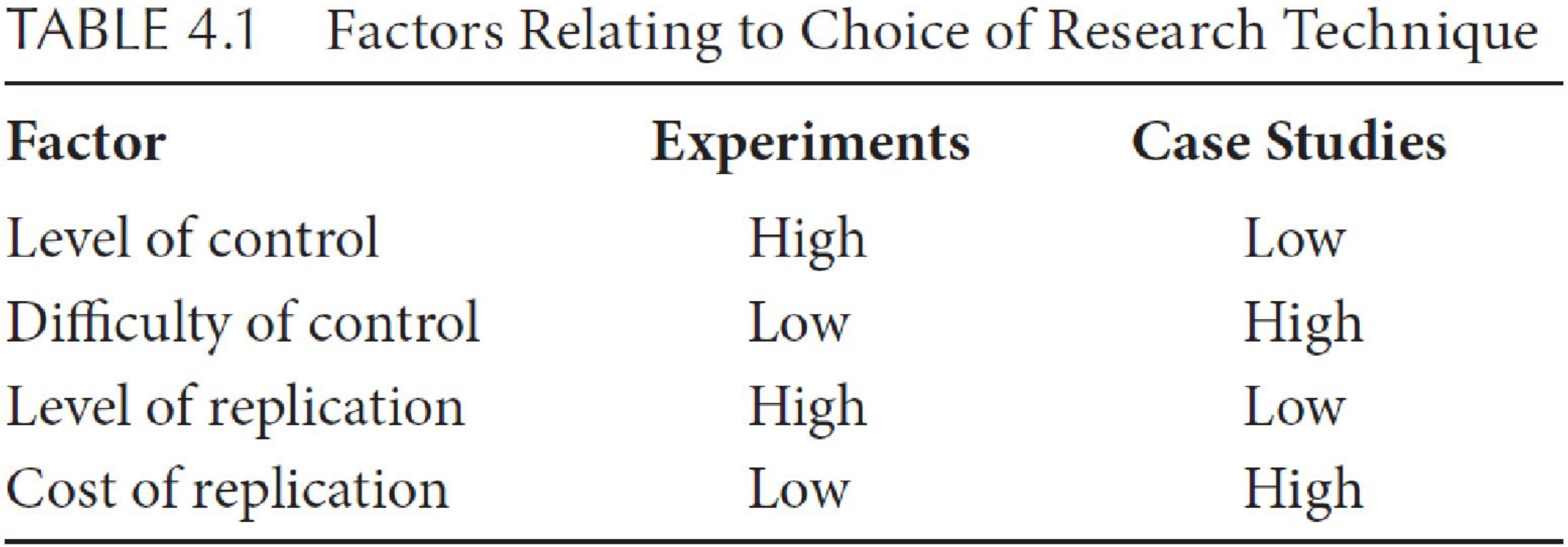


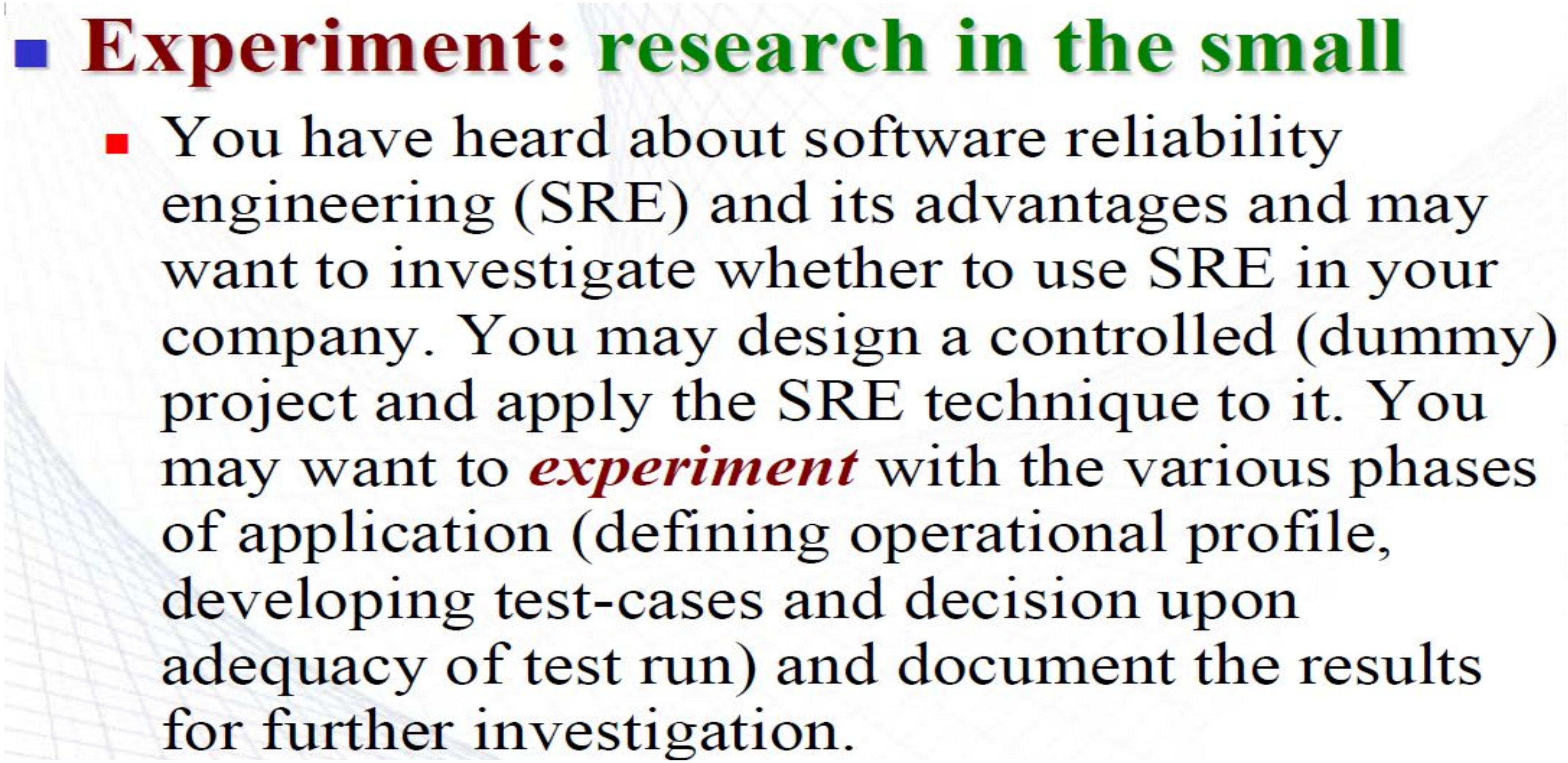
TABLE 4.1 Factors Relating to Choice of Research Technique

Factor	Experiments	Case Studies
Level of control	High	Low
Difficulty of control	Low	High
Level of replication	High	Low
Cost of replication	Low	High

Survey: Research is Large

- ❑The Survey is an empirical method that enables researchers to collect data from a large population.
- ❑The main aim of the survey is to **generalize the findings**
- ❑Surveys try to pool what is happening broadly over large groups of projects
- ❑A survey is a **retrospective study** of a situation to try to document relationships and outcome
- ❑A survey is done after an event has occurred
- ❑Steps of a survey include: ***Identifying the objective, target audience, instruments/questionnaire design, instruments/questionnaire evaluation, analysis, conclusion, and documentation***

Example:



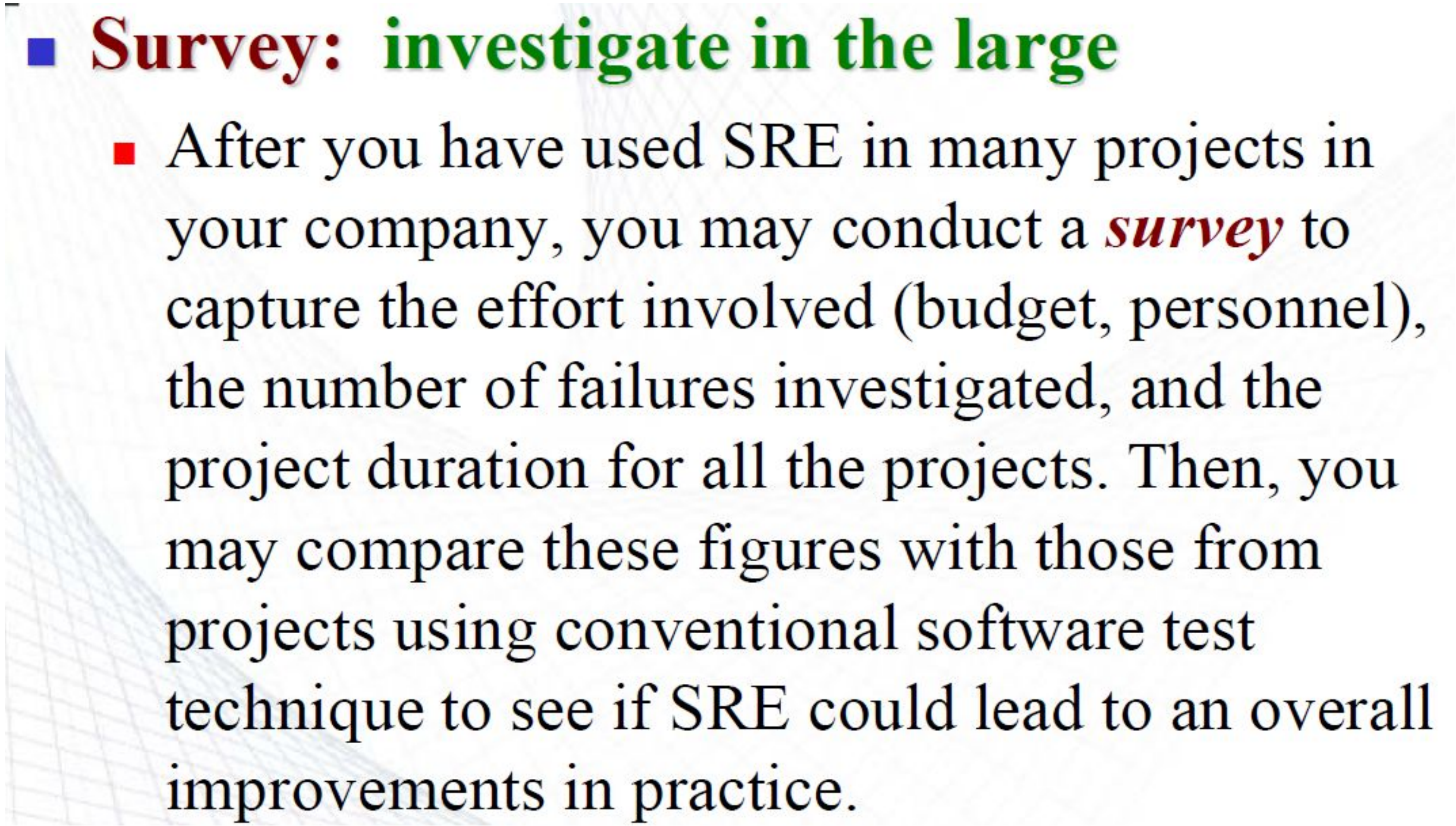
- **Experiment: research in the small**
  - You have heard about software reliability engineering (SRE) and its advantages and may want to investigate whether to use SRE in your company. You may design a controlled (dummy) project and apply the SRE technique to it. You may want to ***experiment*** with the various phases of application (defining operational profile, developing test-cases and decision upon adequacy of test run) and document the results for further investigation.

Example

Case Study: research in the typical

You may want to use or may have used software reliability engineering (SRE) for the first time in a project. You can perform case studies to capture different aspects.

Example



- **Survey: investigate in the large**
  - After you have used SRE in many projects in your company, you may conduct a ***survey*** to capture the effort involved (budget, personnel), the number of failures investigated, and the project duration for all the projects. Then, you may compare these figures with those from projects using conventional software test technique to see if SRE could lead to an overall improvements in practice.

2. Study goal and hypothesis

The goal of the research can be expressed as a hypothesis that you want to test.

Hypothesis: A tentative idea that you think explains the behavior you want to explore

- A Hypotheses should be empirically testable
- The hypotheses must be conceptually clear and specific
- The hypotheses should be related to a body of theory or some theoretical orientation
- **Hypotheses should have quantitative terms and in terms of independent and dependent variables** that are as direct and unambiguous as possible

Hypothesis example

- ***Using the Scrum method produces better quality software than using the XP method***

  - is a hypothesis. However, it is not testable because the notion of quality is not given in a measurable way

You can define “quality” in terms of the defects found and restate the

hypothesis as -

If using the Scrum method produces better quality software than using the XP method then code produced using Scrum will have fewer defects per thousand lines of code than code produced using the XP method.

3. Maintaining Control over Variables

The key discriminator between experiments and case studies is the degree of control over variables. A case study is preferable when you are examining events where relevant behaviors or variables can not be manipulated.

● **State variable** = independent variable that characterizes the project or goals

Example: Suppose your hypothesis involves the effect of programming language on the quality of the resulting code. "Language" is a state variable, and an ideal experiment would involve projects where many different languages would be used

● The **treatment** is any independent variable manipulated by the experimenter

Example: If you want to test whether a new testing strategy is better than other used testing strategies, the new one is a treatment

● After the case studies, you take a sample from the state variables for your project. So, if you have less control over the state variables of your project and if you are not sure about any treatments then you must have to choose a case study instead of an experiment

4. Threats to Validity

No study is perfect. Potential problems with empirical studies are classified as categories of *Threats to Validity*

1. Conclusion Validity: A study has conclusion validity if the results are statistically significant using appropriate statistical tests over the independent and dependent variables.

- Threats to conclusion validity include using the wrong statistical tests, having too small a sample, etc.

2. Construct Validity: Construct validity is used to determine how well a test measures what it is supposed to measure

- For example, using the measure faults per KLOC as a measure of code quality has some threats to validity since its value depends in part on when the measure is taken, for example: during testing(faults found during testing) or after release (faults found by customers)

4. Threats to Validity

3. Internal Validity: Internal validity refers to the cause-effect relationship between independent and dependent variables. A study has internal validity if the treatment actually caused the effect.

- Specific threats include the effects of other, possibly unidentified variables.

4. External Validity: External validity refers to how well you can **generalize from the findings of a study to other situations, people, settings, and measures. In other words, can you apply the findings of your study to a broader context?**.

- seven threats to external validity: **selection bias, history, experimenter effect, Hawthorne effect, testing effect, aptitude-treatment, and situation effect**
- ***Hawthorne effect:*** *The tendency for participants to change their behaviors simply because they know they are being studied.*

5. Human Subject

Consent!!!!!!

Should not be any harm or damage or injury to the subjects while studying.

Software engineering studies rarely involve risks of physical harm to human subjects. The major risks are due to **privacy issues**.