Evidence-Based Software Engineering: The empirical context

Empirical methods are usually defined as qualitative or quantitative. *Quantitative* evaluation uses formal statistical analysis to determine whether outcomes are significantly different from chance. Quantitative studies are used to assess:

- Whether a cause-effect relationship exists. For example, using a randomized experiment to investigate whether a new inspection method can increase the number of faults found when reviewing software requirements documents.
- Whether there are associations between factors. For example, using correlation analysis of observational data to investigate if project success is associated with well-defined requirements.

Qualitative evaluation is concerned with discovering the causes of the behaviour of software developers and managers. For example, investigating what factors motivate or demotivate software developers. Qualitative studies involve:

- Making interpretations based upon explanations provided by developers and managers.
- Recognizing that there may well be different interpretations of a phenomenon.

Types of quantitative study	**Types of qualitative study**
Laboratory experiment: provides a high degree of control but may be difficult to generalise the outcomes. For example, may be based on a simple task, or use of student participants.	*Case study:* an empirical method investigating contemporary phenomena in their context. Information can be obtained from interviews, observations and/or analysis of project records.
Quasi-experiment: involves more limited control by the experimenter but can be used in industry settings, making results more realistic. For example, investigating project productivity before and after major change to a development process.	*Ethnographical study:* where the observer operates within the community but acts in a non-intrusive manner
Opinion Survey: Can provide a broad overview, but may be untrustworthy unless a well-defined population of suitable participants are sampled.	*Interview:* Semi-structured and unstructured interviews are used to gather in-depth views of developers and managers. Qualitative methods such as content analysis and thematic analysis may be needed to analyse the data collected.
Data Mining: Statistical or AI analysis of software project data to investigate relationships between application and component properties such as size and structure and process properties such as developer effort and fault rates.	

Detecting cause-effect relationships

For SE it is rarely possible to just modify the one variable of interest to see what effect this has while keeping all the other variables unchanged. This complicates analysis and may make it difficult to detect small effects. In addition, most experiments in SE have few participants, which means the results of single experiments are untrustworthy, and this problem is rarely avoided by effective use of replication studies.

The *independent* variable(s) (other terms used are ‘stimulus’ or ‘input’) is (are) associated with *cause*. Changes to these arise from the activities of the investigator, and their value should not be affected by the other variables. Using more than one can complicate the analysis of the outcomes.

The *dependent* variable (other terms used include ‘response’ or ‘outcome’ variable) is associated with *effect* and its value should change as a result of changes to the independent variable(s). Measuring it is the means by which the outcomes of the study are captured. We are interested in whether an independent variable causes change to the dependent variable, but our methods are not always rigorous enough to allow us to make cause-effect claims, whereupon we then need additional arguments to support these.

A *confounding factor* is some (undesirable) element in an empirical study that produces an effect that makes it difficult to distinguish between two or more possible causes of an effect (as measured through the dependent variable). For SE two typical examples are the skill levels of the participants in experimental studies and the sampling processes used in surveys.

Threats to validity

Given that there are many factors affecting a study, a key question for its design must be how *trustworthy* the outcomes from it are likely to be? To be adequately *valid*, the results should be valid for the "population of interest" (students, software developers, maintainers,…). A *threat to validity* is therefore a factor that may put the outcomes of a study in question. Three major forms of threat are:

- *internal validity*, is concerned with factors that might have affected the outcomes (the dependent variable) without the researcher's knowledge and which might put the *causal* relationship between treatment and outcome in question – for software engineering this might arise because of the lack of a control group (for example in a before-after investigation of the impact of a process change on productivity, projects undertaken before a change to the process may have been more difficult than those done after the change).
- *external validity* is concerned with how generalisable the results may be to the intended population of interest, here the threat is that the results are only applicable to the particular context of the study – in software engineering these might arise from the selection processes: e.g. using students rather than experienced developers, testers etc
- *construct validity*, is concerned with how well the outcomes of the study are linked to the concepts or theory behind the study – for SE this might be because the model used in the study is inadequate (e.g. comparison between methods to determine which is 'better' without providing a clear definition of what is meant by 'better')

Designing an evaluation study

The *protocol* is a document that describes our plan for conducting a study. It should be developed before the study begins, and any *divergences* from it that occur while performing the study should be formally recorded. Aim is to have a rigorous plan and to address any potential problems in a systematic and consistent way. The protocol should be documented formally (with abstract and references) and should address at least the following elements: *background* (why needed); *context* (where it is to be performed); *detailed form* of the study (processes, activities, tasks etc.); *how* any participants are to be recruited/selected; *data* collection and analysis; *timetable*; *limitations* and constraints.

For a controlled experiment we also need to identify:

- the *independent variables* – those that we can control and change, or influence/measure in some way, and will be determined by the question that we are seeking to answer (since they represent 'cause')
- the *dependent variable* – since this may often not be directly measurable, we might have to identify and use a surrogate (indirect) measure instead

In addition, if we are going to conduct an experiment, we also need to determine the form of *treatment* to be used (how we will manipulate the independent variables in the study). In SE, the treatment might be a process (testing strategy, design method, …) or a product (web browser, development environment, programming language,…)

Many empirical studies in SE involve people, usually referred to as *participants*. Recruiting them:

- involves *ethical issues*—recruiting should not put people under pressure to participate, and as far as possible anonymity should be guaranteed. Protocols for studies involving people usually need to be submitted to some form of ethics committee before they occur
- should aim for a *representative sample* from the domain of interest or a surrogate domain

Data collection can use both 'intrusive' forms (where the participants fill in forms, attend interviews etc.) and also non-intrusive forms where they may be filmed or the computer may maintain a 'log' of their activities. Participants need to be made aware of the use of non-intrusive forms.

For all studies, a really important element is the *dry run*. Studies are not easily repeated if we get things wrong, so we need to check the design, data collection, clarity of instructions etc., by using one or two representative participants who can then provide feedback about these aspects of the study to the experimenters.