Empirical Software Engineering
Introduction & Basic Concepts

Dietmar Winkler

Vienna University of Technology
Institute of Software Technology and Interactive Systems

dietmar.winkler@qse.ifs.tuwien.ac.at
http://qse.ifs.tuwien.ac.at

Motivation

- A major goal in software engineering is the delivery of high-quality software solutions.
- The construction of software products requires professional approaches, e.g., software processes (e.g., Life-Cycle Model, V-Modell XT, Scrum).
- Methods support engineers in constructing and evaluating software products.
  - Constructive approaches, e.g., Model-Driven Development, Test-Driven Development, and pair programming to create new software products.
  - Analytical approaches, e.g., inspection and testing to assess product and process quality.
- Increasing product quality (e.g. less defects), project and process performance (faster delivery of products) requires the application of improved methods and tools.

Questions

→ How can we evaluate and assess improved methods and processes?

→ How can we measure process / product attributes in general?

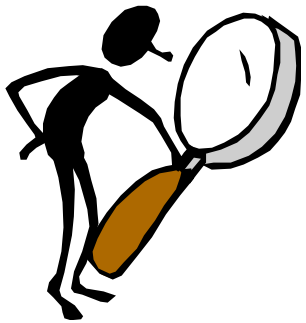
→ How can we conduct an empirical study?

Table of Contents

- Need for empirical studies in the context of software engineering.
- Basic Concepts: Measurement, Collection, Analysis
- Goal Question Metric
- Empirical Strategies: survey, case study, controlled experiment
- Process of an empirical study

Reasons to Conduct Empirical Studies

- New software development technologies come up frequently, e.g. tools, methods
  - Question: Why should we invest in those technologies?
- In other disciplines, technology evaluation is a pre-requisite, …
  … but not in software engineering…
  - Often intuition: “I believe that my method is better than XYZ”?
- Empirical studies in SE are necessary:
  - To prove theories and ‘general wisdom’.
    Example: OO is better
  - To find relationships.
    Example: Relationship between Maintainability and its metrics (e.g. LoC)
  - To choose amongst models/techniques/tools.
    Example: development approaches, inspection reading techniques
  - To judge on the accuracy of models.
    Example: cost models



Need of Empirical Studies

Why should we conduct empirical studies?

- Product evaluation, e.g., prototyping.
- Process evaluation
  - Prototypes are not possible (simulation based on models).
  - A process is just a description until it is used by people.
- Important for research: experimentation is mandatory in other disciplines (e.g., medicine, physics, etc.)
- Experimentation provides a systematic, disciplined, quantifiable and controlled way of evaluating human-based activities.

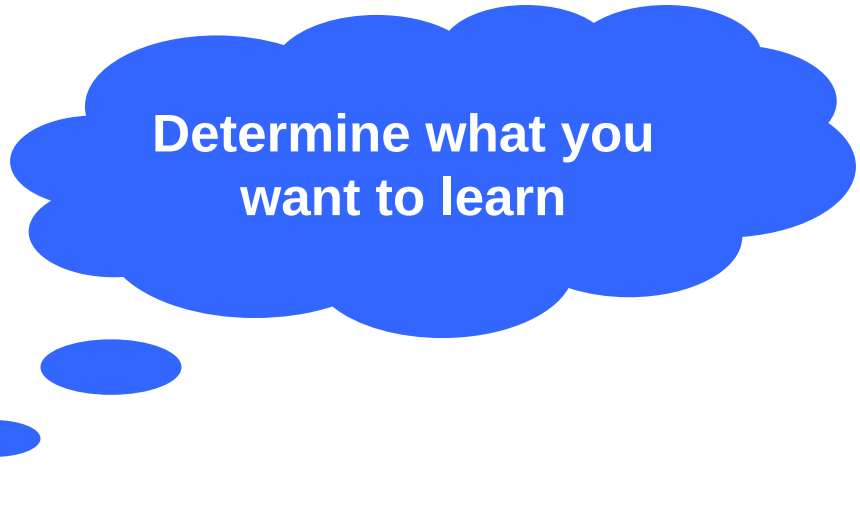
→ You will learn how to select, plan, conduct and analyse an empirical study.

Goals and Benefits

The purpose of a study is

- to explore ...
  - to find out what's happening
  - to seek for new insights
  - to ask questions and to find answers
  - Measurement: usually qualitative
- to describe ...
  - portray accurate profile of situations, events, projects, technologies
  - Measurement: quantitative/qualitative
- to explain ...
  - seek explanation of a situation/problem, usual in the form of causal relationships
  - Measurement: quantitative/qualitative

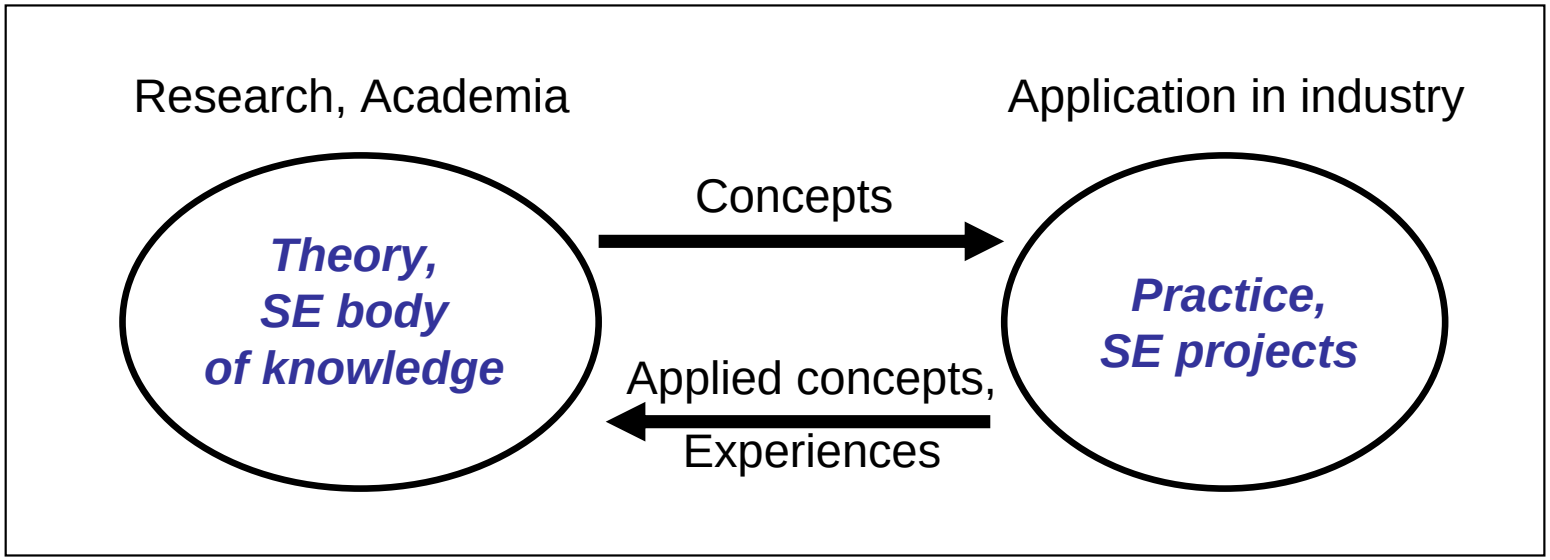
- … relationships, differences, changes



Empirical Studies in SE

Empirical Studies is research to improve Software Engineering Practice.

- Apply theoretical concepts in SE practice.
- Add experiences on their appliance to the SE ‘body of knowledge’
- Improve processes, methods and tools (SPPI approach).
- Verify theories and models.

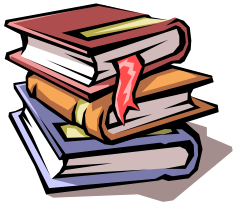


Objects of Empirical Research

Organization(s)

Software processes
e.g. testing process

Empirical Research



Product(s)
e.g. documents, software

Project(s)

Resources
people and tools to conduct process

Table of Contents

- Need for empirical studies in the context of software engineering.
- Basic Concepts: Measurement, Collection, Analysis
- Goal Question Metric
- Empirical Strategies: survey, case study, controlled experiment
- Process of an empirical study

Some Basic Concepts

- Measurement
  - is the process of capturing data which are connected to real-world attributes to describe them.
  - Why is measurement important?
- Data Collection
  - Collection of qualitative / quantitative data according to research questions.
- Data Analysis
  - Analyzing the results according to the research questions; statistical tests.

Measurement

- **Quotes**:
  - “You can’t manage what you can’t measure”, Tom DeMarco
  - “What is not measurable make measurable”, Galileo Galilei

- **Objectives**:
  - One objective of science is to find ways to measure attributes of entities we are interested in.
  - Measurement makes concepts more visible and thus more understandable and controllable.

- **Definition**
  - Measurement is the process by which numbers or symbols are mapped to attributes of entities in the real world in such a way as to describe them according to clearly defined rules.

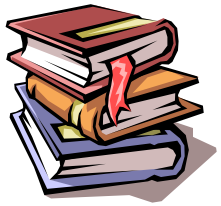
Measurement (Examples)

Process

effort → person days spent from start to end → 10 weeks

- Examples: Development process (V-Modell XT), Testing Process, Inspection, ..

Product



size → No. Lines of Code → 700 LOCs

- Examples: Design Specification (No of pages), Test Suite (number of test cases), Module (LoCs)

Resources

experience → >10 projects → high

- Examples: Project management experience, Testing experience, Design / Architecture experience.

Selected Types of Measures

- Direct vs. Indirect Measures:
  - Direct: obtaining values direct from the study object (e.g., duration, effort)
  - Indirect: calculated values based on various attributes (e.g., efficiency of defect detection = number of defects per time interval)

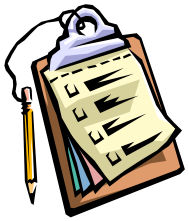
- Objective vs. Subjective Measures:
  - Objective: no judgment in measurement value (e.g., LoC, delivery date)
  - Subjective: reflect judgment of the measurer, depending on the viewpoint
    (e.g., subject defect estimation, questionnaires)

- Quantitative vs. Qualitative data:
  - Quantitative: data expressed as numbers (e.g., data obtained through measurement, statistics)
  - Qualitative: data expressed as word and pictures (e.g., interviews, interpretation)

Data Collection

- Measurement
  focuses on products, processes (typically quantitative data collection)

- Interviews
  based on information obtained from individuals persons or groups (typically qualitative data)



- Questionnaires
  set of questions to obtain information from individuals, e.g., experience, feedback; (typically used in surveys)

- Observation
  selection, recording, and encoding of a set of natural behaviours or other naturally occurring phenomena (typically used in case studies)

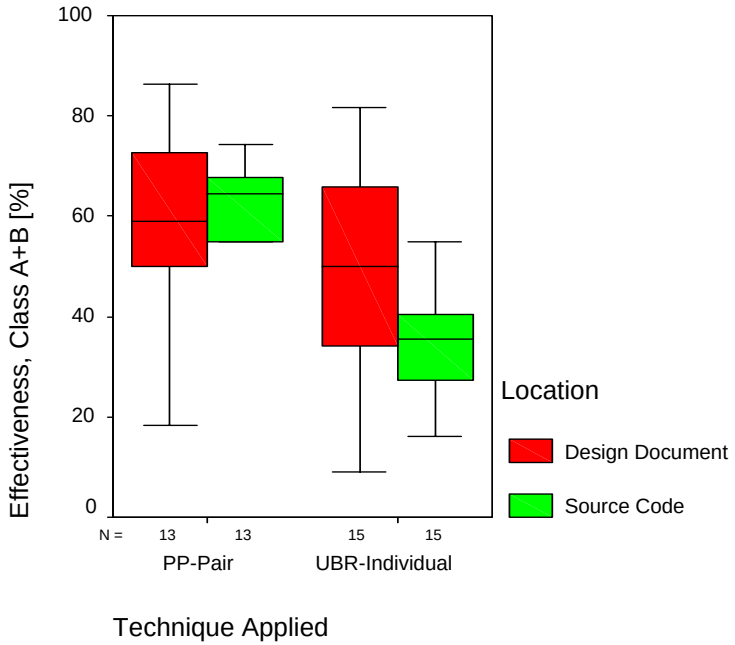
Data Analysis

Purposes of quantitative data analysis

- Describing a population (descriptive statistics)
- Exploring differences between groups (Hypothesis Testing)

Examples:

- Minimum, Mean, Maximum, Standard Deviation.
- Visualization, Statistical Tests to test Hypothesis.



Statistical Tests

	Location	PP-Pair	UBR-Individuals	P-value
Mean	DD+SC	56.3	40.3	0.013 (S)
	DD	56.3	47.3	0.212 (-)
	SC	56.3	35.3	0.004 (S)
Std.Dev	DD+SC	20.6	13.6	-
	DD	26.7	20.6	-
	SC	17.9	11.4	-

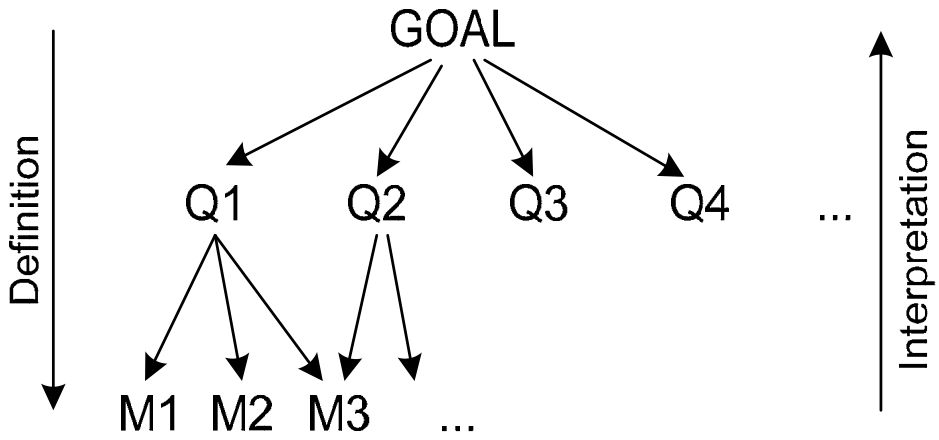
Table of Contents

- Need for empirical studies in the context of software engineering.
- Basic Concepts: Measurement, Collection, Analysis
- Goal Question Metric
- Empirical Strategies: survey, case study, controlled experiment
- Process of an empirical study

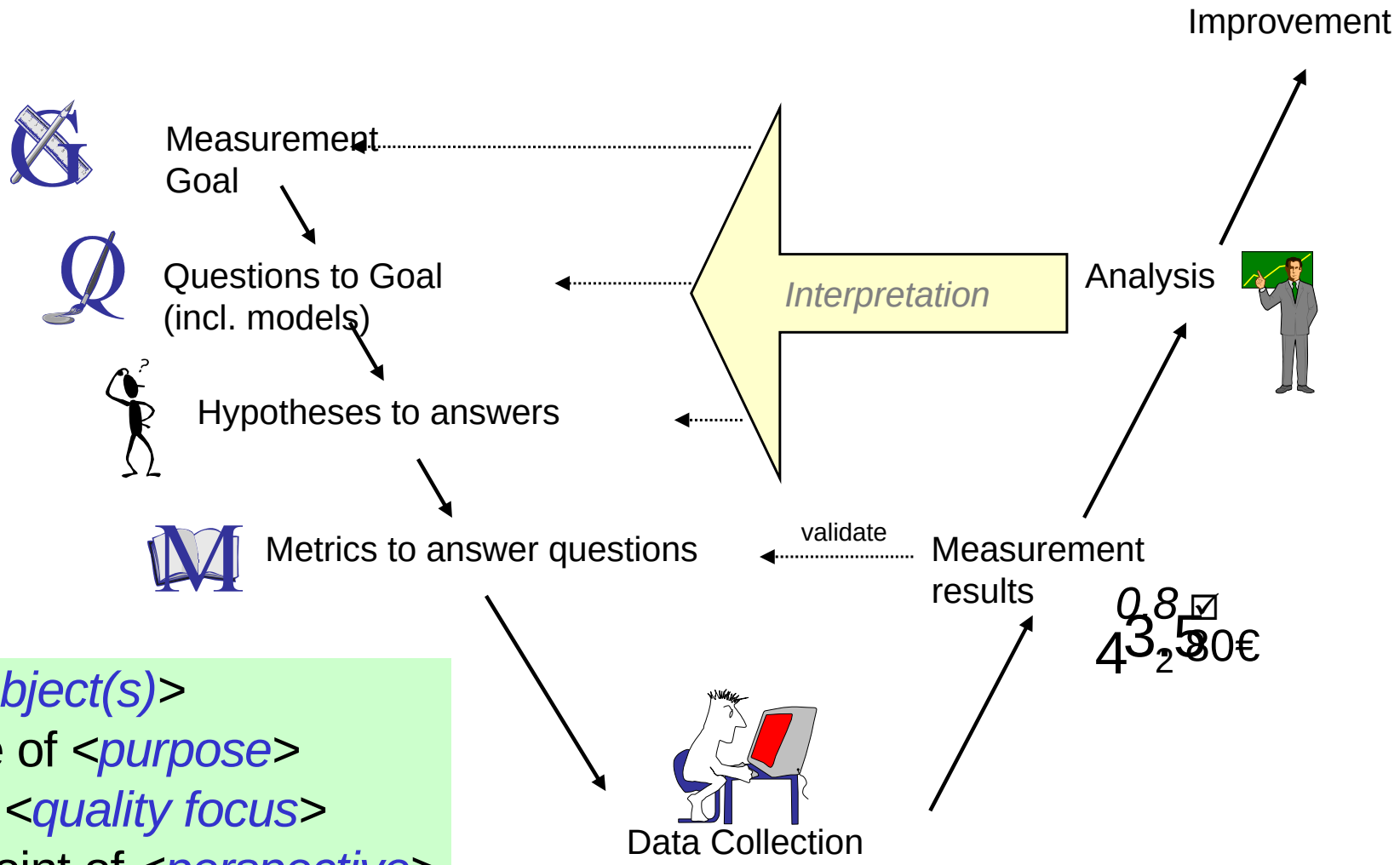
Goal Question Metrics (GQM)

- A framework for systematic measurement, data collection and analysis.
- The main goal of GQM is to provide a framework for the measurement of software quality [Basili et al, 2000]

- GQM consists of three major levels:
  - Conceptual level (GOAL)
    Objects of measurement are products, processes, and resources
  - Operative Level (QUESTION)
    characterization of questions according to the goals.
  - Quantitative Level (METRIC)
    Measures to answer defined questions.



GQM Methodology

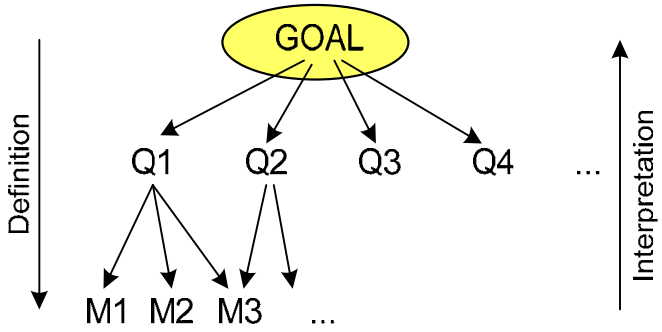


Analyze the <*object(s)*>
for the purpose of <*purpose*>
with respect to <*quality focus*>
from the viewpoint of <*perspective*>
in the context of <*context*>

GQM-Goal Template

Example: GQM Goals

- The management wants to optimize software project development costs.
- Required information: Project effort / duration within individual development phases for selected projects.
- How can we achieve this information?



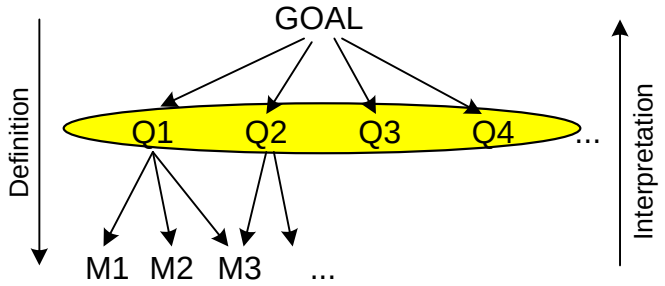
Step 1: Derive goal (based on the GQM template):

→ Characterization of software development costs in a certain project.

- Analyze the software development process (object)
- for the purpose of characterization (purpose)
- with respect to effort (quality focus)
- from the viewpoint of the management (perspective)
- in the context project A (context)

Example: Required information to achieve the goal

Step 2: Derive appropriate questions



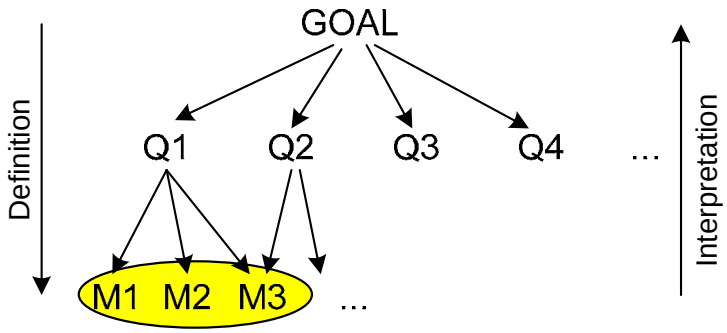
- What information is required to achieve the measurement goal?

- Some possible questions:
  - What is the distribution of effort among development phases/activities?
  - Define the typical effort per module / component (comparability of the complexity of individual modules / components)
  - Define the effort spent on rework (in case of defects)
  - How many hours are spent on documentation?
  - …

- A set of questions operationally define the goal.
- A set of metrics provide answers to every question.

Example: Metrics to describe process behavior

- Metrics to measure product / project attributes regarding the goal of the investigation.



- Examples:
- What is the distribution of effort among development phases/activities?
  - Person months for every phase (e.g., 5 person months for requirements elicitation; 4 person months for design …)
  - Duration in months, i.e., 1 months for requirements involving 5 persons.

- Define the typical effort per module / component (comparability of the complexity of individual modules / components)
  - Depending on the architecture and design
  - Consider different levels of systems complexity.
- Number of defects found during review cycles (rework / QA effort)
  - Decision to include additional analytical quality assurance steps.

Table of Contents

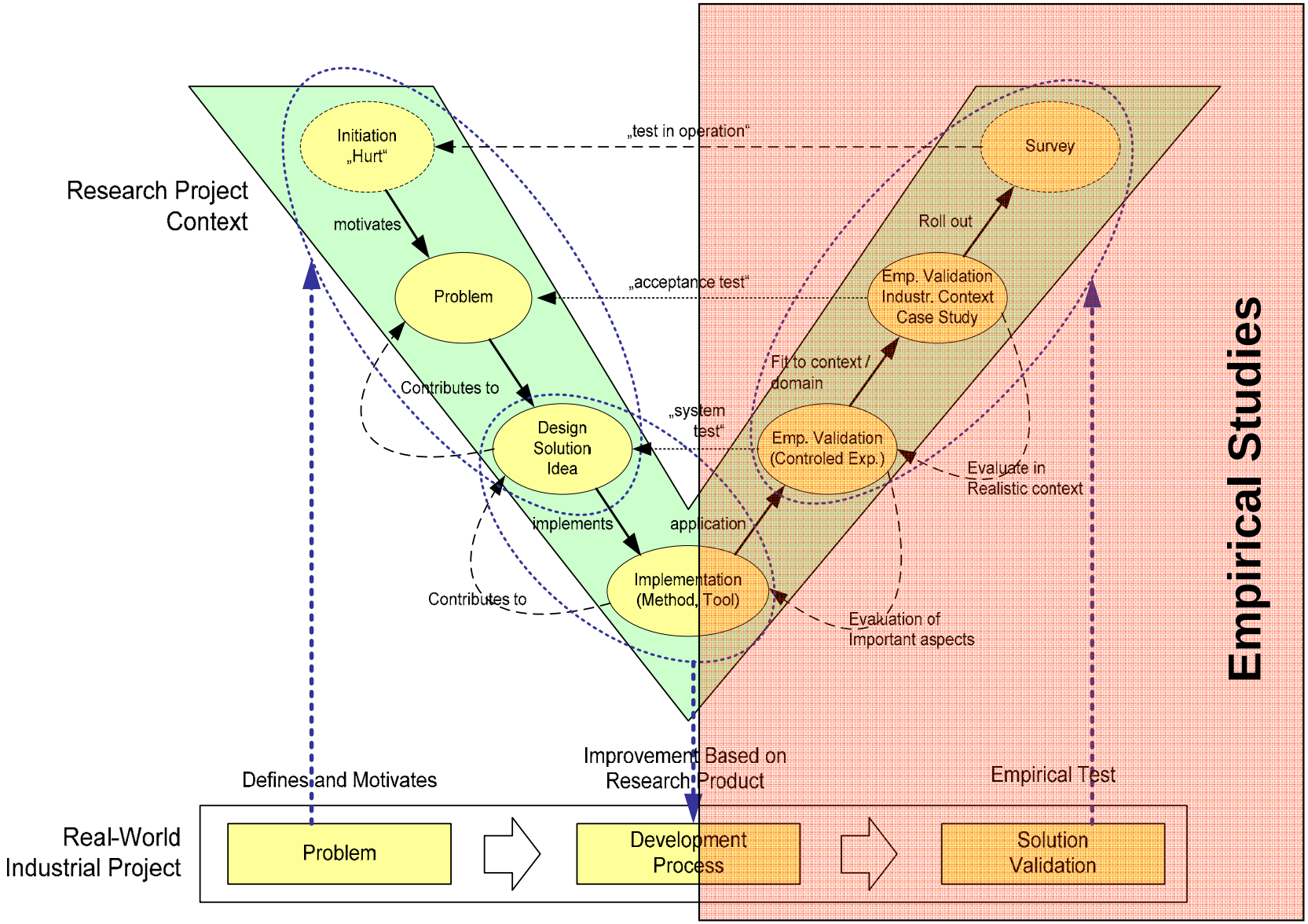
- Need for empirical studies in the context of software engineering.
- Basic Concepts: Measurement, Collection, Analysis
- Goal Question Metric
- Empirical Strategies: survey, case study, controlled experiment
- Process of an empirical study

Introduction

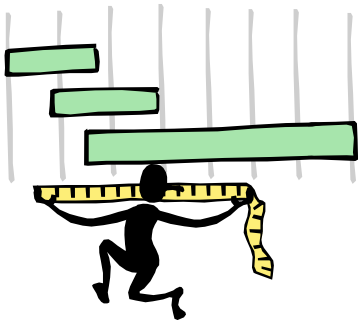
- Empirical studies provide a systematic, disciplined, quantifiable and controlled way of evaluating human-based activities.
- Empirical studies are important for scientific work to get knowledge of products, processes and resources (V-model).
- Empirical methods are important techniques for software quality improvement.
- Different study strategies aim at focusing on individual steps of product / process progress (e.g., laboratory evaluation and simulation, organization case studies, cross-company surveys etc.)

- You will learn
  - V-Model of empirical software engineering.
  - Different strategies of empirical research (survey, case study, controlled experiment)

Empirical Studies in the Context of the V-Model



Different Empirical Strategies



Controlled Experiments

- Measuring the effects of one or more variable(s) on other variable(s)
- Detailed investigation in controlled conditions (relevant variables can be manipulated directly, precisely and systematically)

Case Studies

- Development of detailed, intensive knowledge about a single case or of a small number of related cases
- Detailed investigation in typical conditions

Surveys

- Collection of information in standardized form from groups of people or projects.
- Usually performed retrospectively.
- The use of a technique/tool has already taken place, relationships and outcomes should be documented.

Empirical Strategies

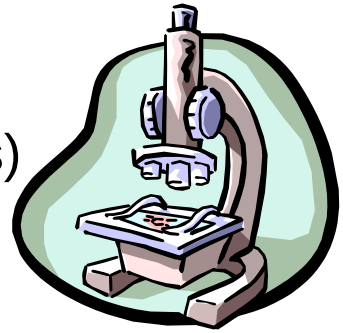
- Controlled Experiment:
  - laboratory environment.
  - an operation is carried out under controlled conditions.
  - manipulate one or more variables and keep all other variables at fixed levels.
- Case Study:
  - Monitoring projects or activities.
  - data collection for a specific purpose.
  - observational study.
- Survey:
  - investigation performed in retrospect.
  - interviews and questionnaires.

Strategy	**Quantitative** (data expressed as numbers)	**Qualitative** (data expressed as words or pictures)	**Study Effort** (always depends on context and research topic)
Experiment	X		(very) high
Case Study	X	X	Medium
Survey	X	X	Low/Medium

Controlled Experiment: Fact Sheet

Purpose:

- Measuring the effects of one or more variable(s) on other variable(s)
- Detailed investigation in controlled conditions (relevant variables can be manipulated directly, precisely and systematically)

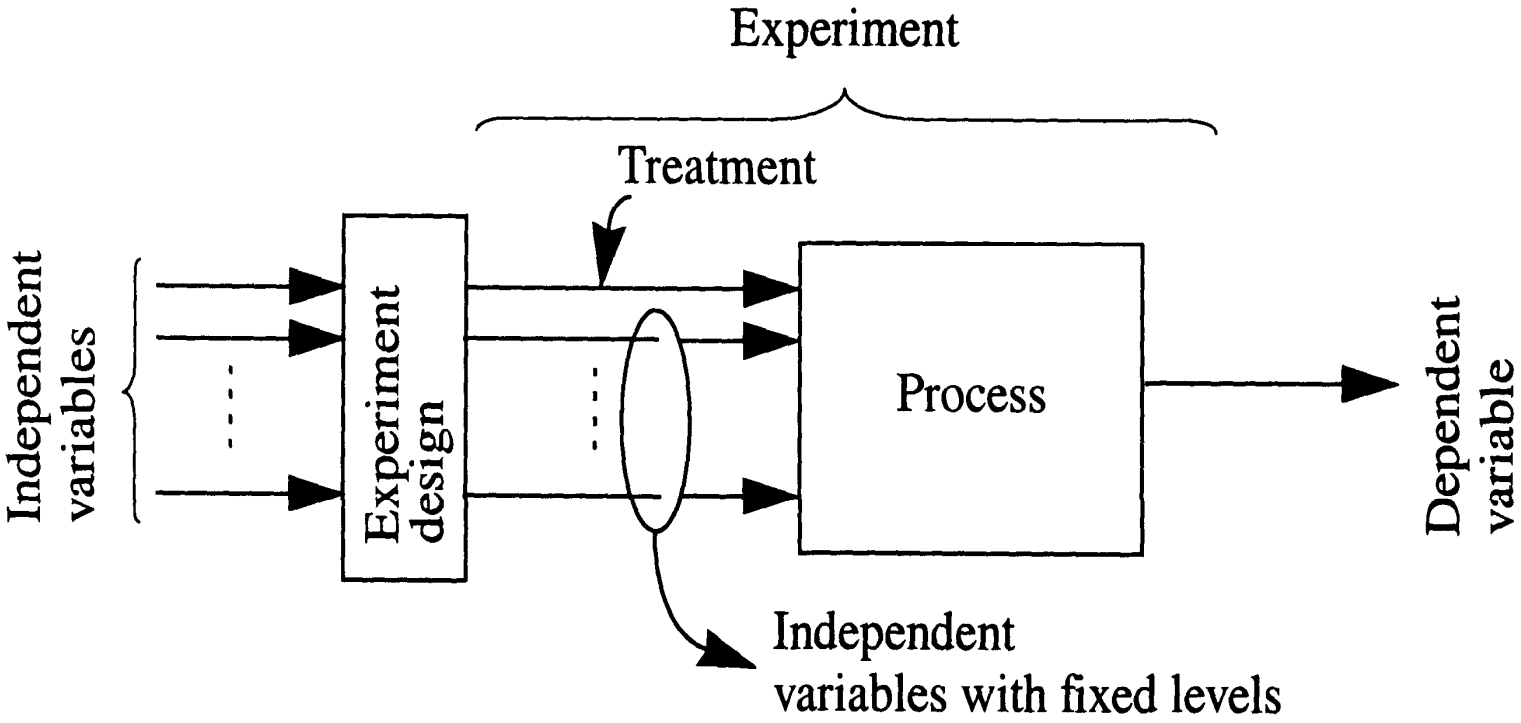


When select an experiment?

- When appropriate: control on who is using which technology, when, where and under which conditions.
- Level of control: high
- Data collection: process and product measurement, questionnaires
- Data analysis: statistics, compare of groups, treatments, etc.

- Pro's: help establishing causal relationships, confirm theories.
- Con's: representative? Challenging to plan in a real-world environment. Application in industrial context requires compromises.

An Experiment

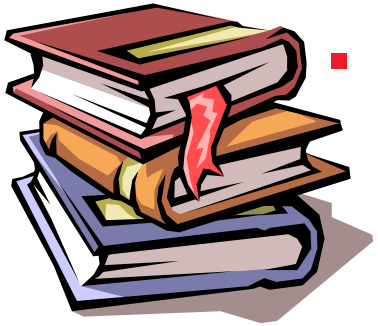


Wohlin et al., Experimentation in Software Engineering, Kluwer Academic Publishers, 2000.

Experimentation Vocabulary

- Dependent Variable: studied to see the effect of changes in the independent variables (e.g. productivity of development team)
- Independent Variable: all variables that are manipulated and controlled (e.g. new development method, experience of developers)
  - Factor: set of investigated independent variables.
  - Treatment: one particular value of a factor (e.g. new and old development method)
- Objects (e.g. documents) and subjects (i.e. participants)

Case Study: Definition



- Empirical study to objectively investigate a new technology in a somewhat realistic setting.
- Method for research evaluation by monitoring a project and collecting data over time. Data collection is derived from a specific goal for the project.
  A certain attribute (e.g., reliability, cost) is monitored and data is collected to measure that attribute. [Zelkowitz et al., 1998].
- Trial use of a technology on a full-scale project, with the objective of comparing the effect of the new technology with that of other technologies or current practice.

Case Study: Fact Sheet

Purpose:

- Development of detailed, intensive knowledge about a single case or of a small number of related cases.
- Detailed investigation in typical conditions.

When select a Case Study?

- When appropriate: change (new technology) is wide-ranging throughout the development process, want to assess a change in a typical situation
- Level of control: medium
- Data collection: product and process measurement, questionnaires, interviews
- Data analysis: compare case study results to a baseline (sister project, company baseline)

- Pro's: applicable to real world projects, help answering why and how questions, provide qualitative insight
- Con's: difficult to implement a case study design, analysis of results is subjective

Survey: Fact Sheet

Purpose:

- A retrospective study of a situation to try to document relationships outcomes.

When select a survey?

- When appropriate: for early exploratory analysis. Technology change implemented across a large number of projects, description of results, influence factors.
- Level of control: low
- Data collection: questionnaires, interviews
- Data analysis: comparing different populations among respondents, association and trend analysis, consistency of scores.

- Pro's: generalization of results is usually easier (than case study), applicable in practice.
- Con's: little control of variables, questionnaire design is difficult (validity, reliability), execution is often time consuming (interviews).

Survey Types in Software Engineering

State-of-the-art Surveys

- Ask people on state-of-the-practice, best practices.
  - Inside an organization: people, departments, business units
  - Over organizations: people with a specific function (e.g. QA, engineer), people in specific departments.

Literature Surveys

- Analyze existing literature (papers, books, notes) to determine the state-of-the-art, best practices on a topic.

Trend Surveys

- Evaluate demand of particular products or services and predict their future.
  - Conducted by institutes like Ovum, Gartner & IDC.
  - Also by asking people in organisation.

Selecting an Empirical Strategy

How to select the appropriate strategy for a study:

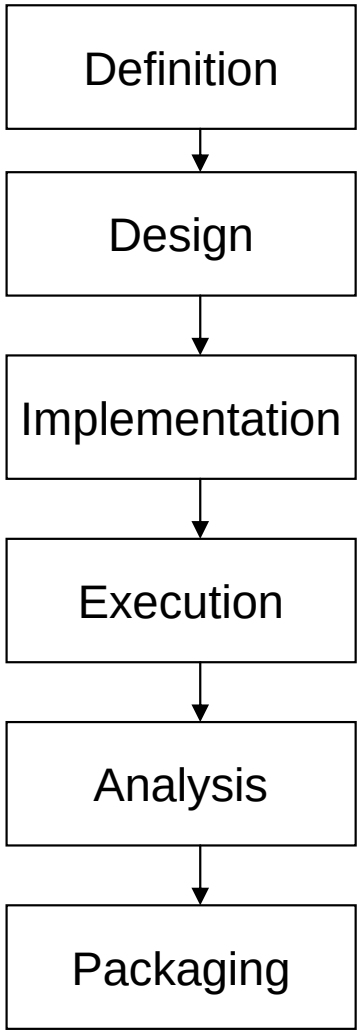
- Purpose of study
  - Exploratory, descriptive or confirmatory
  - Questions concerning what, how, how many, where, for whom
- Degree of control
  - Possibility to ‘arrange’ the real world
  - Required versus possible degree of control
- Cost
  - The relative costs for doing a study;
    e.g. costs for doing experiments are considered as being high
- Risk
  - Probability that study might fail and its consequence

Table of Contents

- Need for empirical studies in the context of software engineering.
- Basic Concepts: Measurement, Collection, Analysis
- Goal Question Metric
- Empirical Strategies: survey, case study, controlled experiment
- Process of an empirical study

„High-Level“ Experiment Process

An overview on the high level process

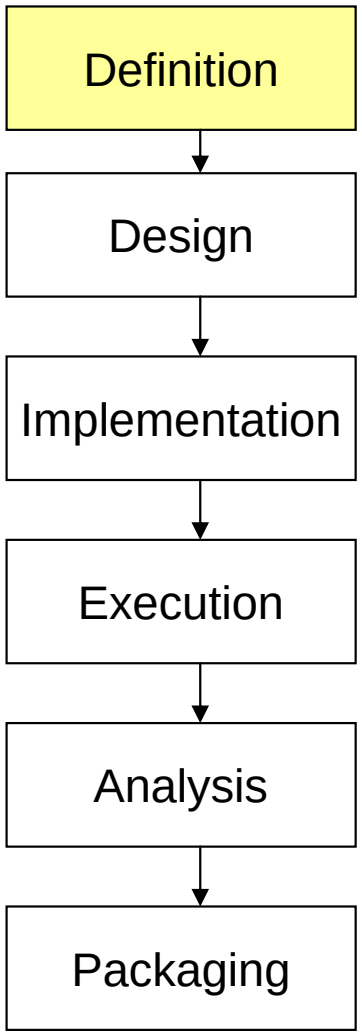


- Definition: Determine study goal(s) and research hypothesis(es). Select type of empirical study to be employed.
- Design: Operationalize study goal(s) and hypotheses. Make study plan: what needs to be done by whom and when.
- Implementation: Prepare material required to conduct the study.
- Execution: Run study according to plan and collect required data.
- Analysis: Analyze collected data to answer operationalized study goal and hypotheses
- Packaging: Report your study so that external parties are able to understand results and context of the study.

Research Proposal: Content

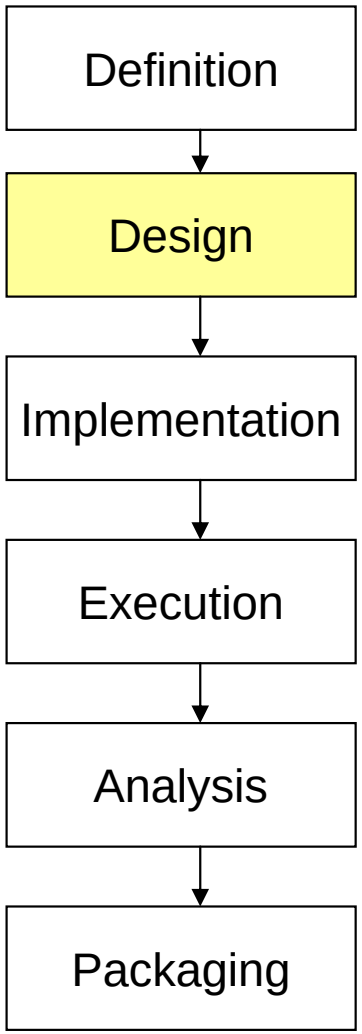
1. **Introduction and motivation**
   - why is the research relevant.
   - description of issues or points.
2. **Relevant prior work**
   - what is the work based on.
   - what are the other relevant research results.
   - what is the "research gap" that this research contributes to.
   - it is sufficient to refer to main relevant work.
3. **Research Objectives, questions and hypotheses**
   - explicit articulation of the research objectives (higher level goals for the research)
   - explicit definition of the research hypotheses and questions (more specific statement)
4. **Empirical study design and arrangements**
   - overall design of the study.
   - description of study arrangements.
   - description data collection procedures and protocols.
5. **Definition of metrics**
   - definition of metrics used in the study, include a list and definition of most important metrics.
6. **Data analysis methods**
   - description of the methods and techniques used in data analysis.
7. **Validity threats and control**
   - description of potential threats and how they will be mitigated
   - how generalizeable the results are?

Experiment Process: Definition



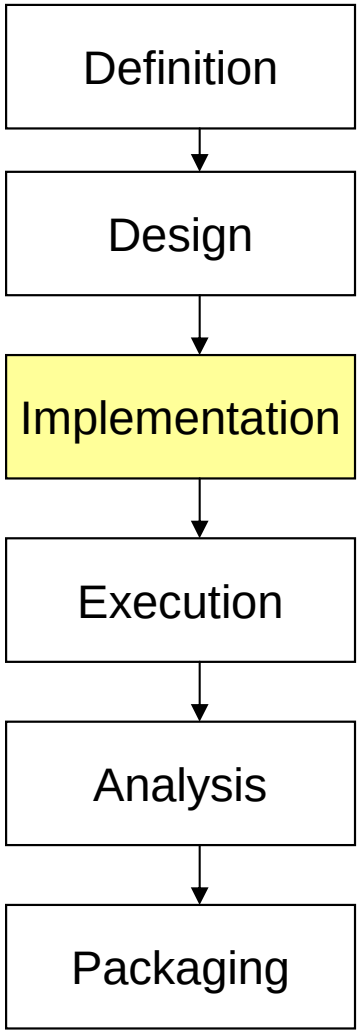
- Determine study goal(s) and research hypothesis(es). Select type of empirical study to be employed.
- Define Research Objectives:
  - explicit articulation of the research objectives (higher level goals for the research)
  - Example: the new model will improve software development productivity.
- Define Hypotheses:
  - explicit definition of the research hypotheses and questions (more specific)
  - Example: Method 1 performs better than method 2, because …

Experiment Process: Design



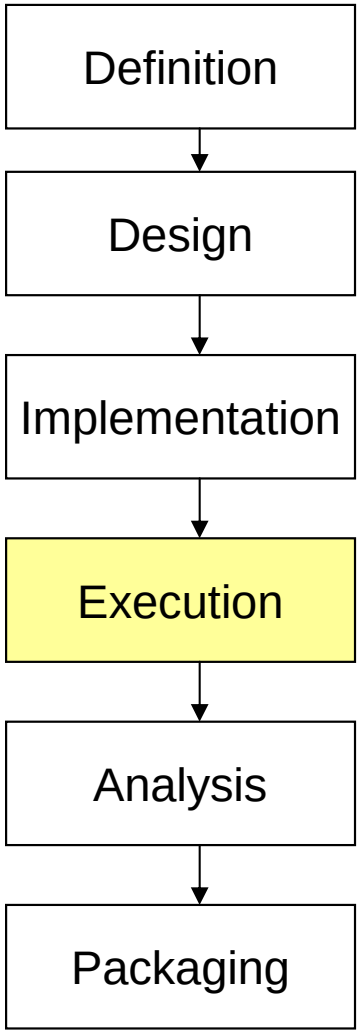
- Operationalize study goal(s) and hypothesis(es).
  Make study plan: what needs to be done by whom and when.
- Determine what needs to be observed / measured;
  quantitative and qualitative data.
- Maximize validity of results;
  identify what effects might influence my findings.
- Maximize reliability of the study (to enable replication)
  → documentation of procedures, context, measurements.

Experiment Process: Implementation



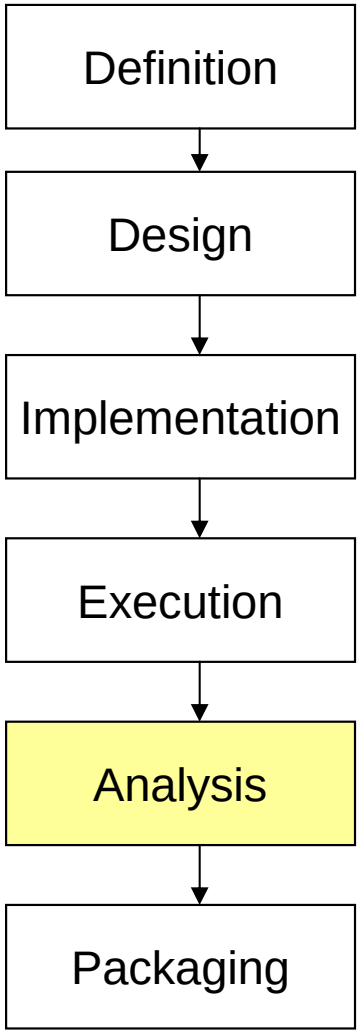
- Prepare material required to conduct the study.
- Use intensive reviews to check the experiment material for correctness.
- Apply Pilot-Tests to verify / improve the experiment material.
  - Are instructions clear, understandable, consistent?
  - Are tasks too simple or too difficult?
  - Can all data be collected as intended?
  - Is the schedule appropriately planned?
  - Note: participants in pilot-tests should be representative for subjects.
- Example:
  - We conducted a pilot study (including a smaller number of participants) with similar material to verify and improve the experiment package.

Experiment Process: Execution



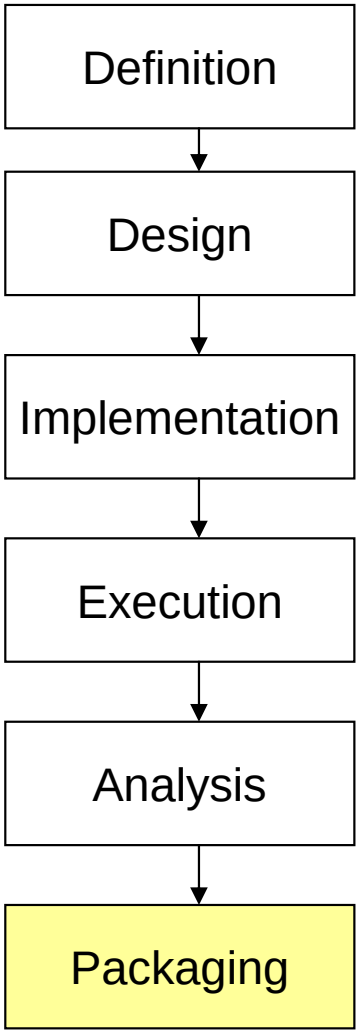
- Run study according to plan and collect required data.
- Example:
  - Paper-based data collection (during the experiment)
  - Separated data submission session using a web-tool.

Experiment Process: Analysis



- Analyze collected data to answer operationalized study goal and hypotheses.
- Basic Steps:
  - Data collection
  - Check data for consistency and credibility
  - Create descriptive statistics and visualize data
  - Perform statistical analysis / comparison
  - Interpret results.
- Data validation ensures the correctness and completeness of collected data. Consider …
  - exceptionally high/low values, Null Values
  - Missing Values, Missing Records
  - Inconsistent values

Experiment Process: Packaging & Publication



- Report your study so that external parties are able to understand results and context of the study.
- Report your study to be replicated by others.

Summary

- Experimentation provides a systematic, disciplined, quantifiable and controlled way of evaluating human-based activities.
- The purpose of a study is to explore, to describe, and to explain relationships, differences, changes of products, processes, and resources.
- Measurement provides quantitative and qualitative data of the study object.
- Data collection approaches are basic elements of empirical studies (e.g. measurement, interviews, questionnaires, observation).
- Data analysis describes data of the study, relationships between different entities, etc. Statistical tests are used to falsify hypothesis.
- Goal question metric (GQM) approach is a framework for the measurement of software quality.
- Main study strategies are controlled experiments, case studies, and surveys.
- A study consists of a defined sequence of steps (from definition of the initial study to packaging and reporting of study results).

References

- V. Basili, G. Caldiera, D. Rombach: „The Goal Question Metric Approach“, 2000.
- B. Boehm, H.D. Rombach, M.V. Zelkowitz: „Foundations of Empirical Software Engineering – The Legacy of Victor R. Basili“, Springer, 2005.
- Freimut et al.: "Empirical Studies in Software Engineering", Tutorial, VISEK Technical Report, 2002.
- IESE Tutorials on Empirical Software Engineering.
- A. Jedlitschka and D. Pfahl: "Reporting Guidelines for Controlled Experiments in Software Engineering", ISESE, 2005.
- B. Kitchenham: “Evidence-Based Software Engineering and Systematic Literature Review”, Profes, 2006.
- C. Wohlin, P. Runeson, M. Höst, M.C. Ohlsson, B. Regnell and A. Wesslen: "Introduction to Experimentation in Software Engineering", Kluver, 2000.
- M. V. Zelkowitz, D. R. Wallace: “Experimental Models for Validating Technology”, IEEE Computer, 1997.

Thank you for your attention

Contact:

Dipl.-Ing. Dietmar Winkler

Vienna University of Technology
Institute of Software Technology and Interactive Systems
Favoritenstr. 9-11/188, A-1040 Vienna, Austria

dietmar.winkler@qse.ifs.tuwien.ac.at
http://qse.ifs.tuwien.ac.at

This research work has been supported by a Marie Curie Transfer of Knowledge Fellowship of the European Community's 6th Framework Programme under the contract MTKD-CT-2005-029755: CzechVMXT.