BLEKINGE INSTITUTE OF TECHNOLOGY

BTH

Software Metrics (PA1407)

Lecture 4

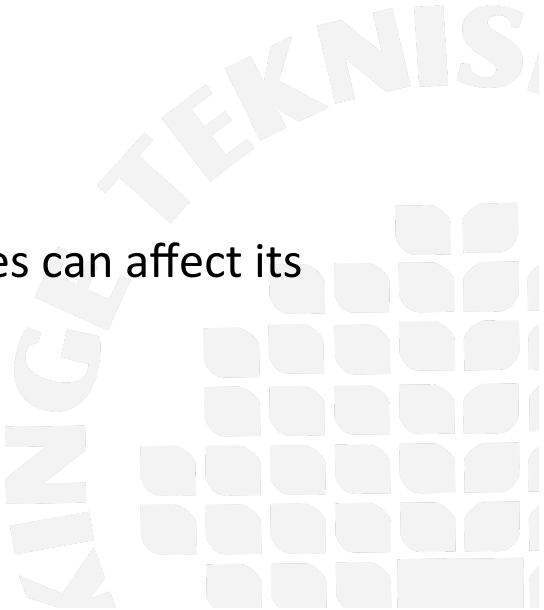
Empirical Studies

Software metrics data collection and analysis techniques

BLEKINGE INSTITUTE OF TECHNOLOGY
BTH

Principles of empirical studies

- Study type and control over variables
  - Survey
  - Case study
  - Experiment
- Study goals/hypothesis
  - What do you want to investigate?
- Maintaining control over variables
  - Once you have a hypothesis, you must decide what variables can affect its truth? And how much control you have on these variavbles
- Threats to validity
  - Internal, external, construct and conclusion validity.



BLEKINGE INSTITUTE OF TECHNOLOGY
BTH

Study type and control on variables

- Survey is a retrospective study of a situation to try to document relationships and outcomes.
  - You have no control over the situation at hand.
- Both case studies and formal experiments are usually not retrospective.
  - You decide in advance what you want to investigate and then plan how to capture data to support your investigation.
- Controlled experiments involve testing of well defined hypotheses concerning postulated effects of independent variables on dependent variables in a setting that minimizes other factors that might affect the outcome.
  - This setting, however, is not possible/realistic in many cases.

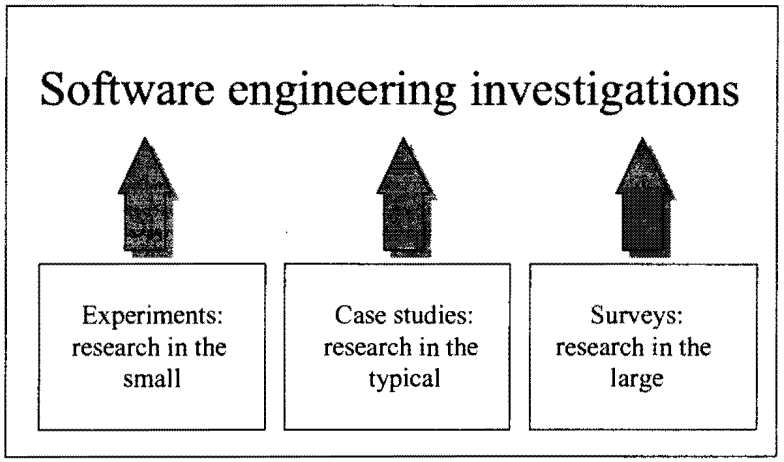
Study type and control on variables

- Empirical studies that involve observations where potential confounding factors can not be controlled and/or subjects can not be assigned to treatment or control groups are called observational studies and/or quasi experiments.
- A case study is a quasi experiment where you identify key factors that may affect the outcome of an activity and then document the activity: its inputs, constraints, resources, and outputs.

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

Study type and control on variables

- Differences among these study types /research methods are also reflected in their scale.



BLEKINGE INSTITUTE OF TECHNOLOGY
BTH

Study goals and hypothesis

- Goal (s) helps you to decide which type of research method is most appropriate for your situation.
- The goal can be expressed as hypothesis
- The hypothesis is the tentative idea that you think explains the behavior you want to explore
  - Using Scrum produces better quality software than using the extreme programming (XP).
    - Examine past records to assess what happened when a particular group used each method (a survey)
    - Evaluate activities of an organization during or after using scrum (a case study)
    - Conduct a carefully controlled comparison of those using Scrum with those using XP (a formal experiment).
  - In all cases you are testing to see if data you collect will confirm or refute the hypothesis you have stated.

BLEKINGE INSTITUTE OF TECHNOLOGY
BTH

Maintaining control over variables

- What variables can affect truth of stated hypothesis?
- You have to see how much control you have over these variables?
- A case study is preferable when you are examining events where relevant behaviors cannot be manipulated
  - Example: If you are investigating the effect of a design method on the quality of the resulting software, but you have no control over who is using which design method, then you probably want to do a case study.
- Experiments are done when you can manipulate behavior.
  - If you can control which subjects use Scrum, and which subjects use XP, and when and where they are used, then an experiment is possible.
    - This type of manipulation can be done in “toy” situations.

BLEKINGE INSTITUTE OF TECHNOLOGY
BTH

Planning experiments and case studies

- Process phases
  - Conception
  - Design
  - Preparation
  - Execution
  - Analysis
  - Decision making

BLEKINGE INSTITUTE OF TECHNOLOGY
BTH

Relevant and meaningful studies

- Confirming theories and "conventional wisdom"
  - Confirming a common notion that pair programming increases code quality
    - What type of study should be conducted to test this idea/hypothesis?
- Exploring relationships
  - How does the design structure affect the maintainability of the code?
    - How to explore this relationship?
- Evaluating the accuracy of models
- Validating measures
  - A measure is said to be "valid" if it reflects the characteristics of an attribute under differing conditions.
    - Several measures claim to measure the complexity of code. How to validate which one is better or more realistic?

BLEKINGE INSTITUTE OF TECHNOLOGY
BTH

Software metrics data collection

"Data should be collected with a clear purpose in mind. Not only a clear purpose but a clear idea as to the precise way in which they will be analysed so as to yield the desired information. It is astonishing that men, who in other respects are clear- sighted, will collect absolute hotchpotches of data in the blithe and uncritical belief that analysis can get something out of it."

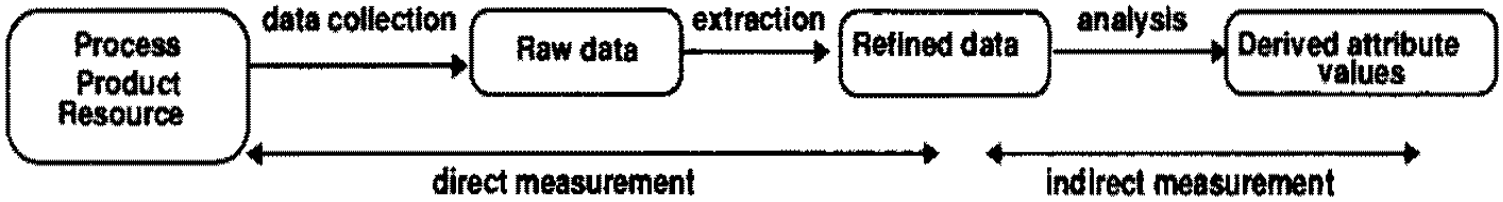
(Moroney, 1950)

BLEKINGE INSTITUTE OF TECHNOLOGY
BTH

What is good data

- Are they correct?
- Are they accurate?
- Are they appropriately precise?
- Are they consistent?
- Are they associated with a particular activity or time period?
- Can they be replicated?

How to define the data



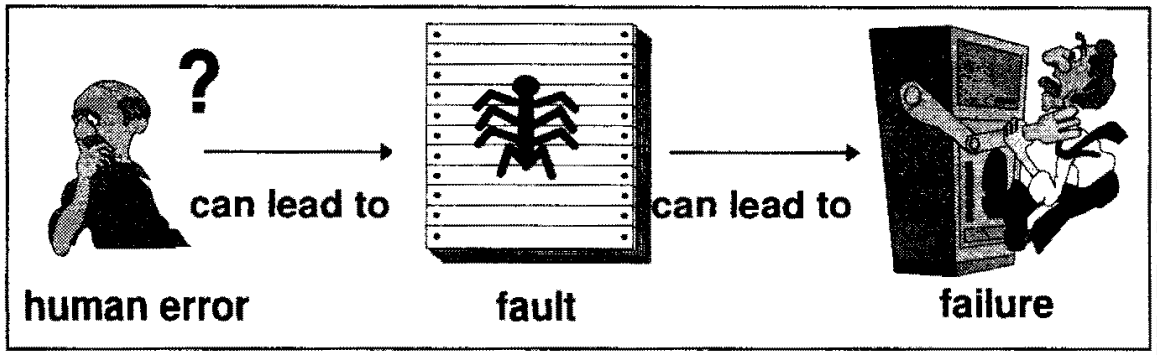
BLEKINGE INSTITUTE OF TECHNOLOGY
BTH

Data collection example: Software Quality

- No software developer consistently produces perfect software the first time. It is important to measure aspects of software quality
  - How many problems have been found with a product?
  - Whether the product is ready for release to the next development stage or to the customer?
  - How the current version of a product compares in quality with previous or competing versions?
  - How effective are the prevention, detection, and removal processes?
- The terminology used to support this investigation must be precise, allowing us to understand the causes as well as the effects of quality assessment and improvement efforts.

Software Quality

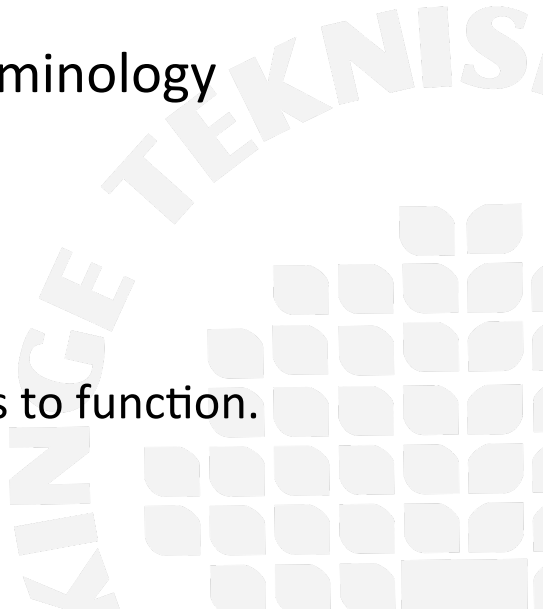
- Terminology – IEEE standard 729
  - A fault occurs when a human error results in a mistake in some software product. (incorrect code as well as incorrect instructions in the user manual)
  - A failure is the departure of a system from its required behavior.



BLEKINGE INSTITUTE OF TECHNOLOGY
BTH

Problem with problems

- The terminology used to describe software problems is not uniform.
  - If an organization measures its software quality in terms of faults per thousand lines of code, it may be impossible to compare the result with the competition if the meaning of "fault" is not the same.
- Researchers and practitioners differ in their use of terminology
  - Errors - often means faults
  - Anomalies
  - Defects – normally refers collectively to faults and failures
  - Bugs – faults occurring in the code
  - Crashes – a special type of failure, where the system ceases to function.



BLEKINGE INSTITUTE OF TECHNOLOGY
BTH

The problem with problems

- Since the terminology varies widely, it is important for you to define your terms clearly, so that they are understood by all concerned.
- Whenever a problem is observed, record following key elements
  - Location - where did the problem occur
  - Timing - when did it occur
  - Symptom - what was observed
  - End Result - which consequences resulted
  - Mechanism - how did it occur
  - Cause - why did it occur
  - Severity - how much was the user affected
  - Cost - how much did it cost

BLEKINGE INSTITUTE OF TECHNOLOGY
BTH

Failure report template

Failure report

Location: such as installation where failure was observed

Timing: CPU time, clock time or some temporal measure

Symptom: type of error message or indication of failure

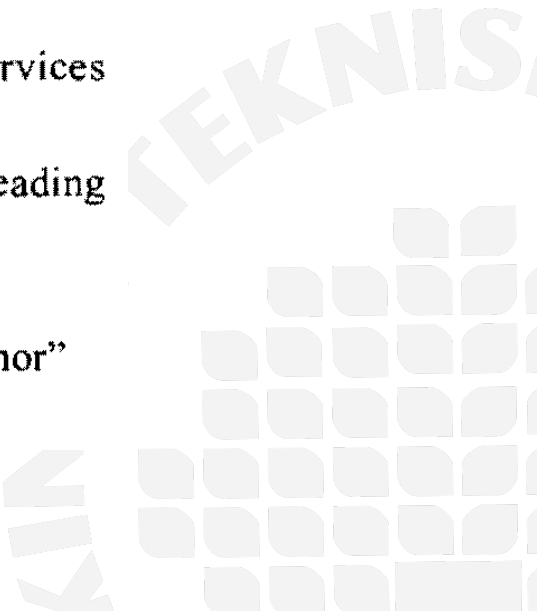
End result: description of failure, such as "operating system crash," "services degraded," "loss of data," "wrong output," "no output"

Mechanism: chain of events, including keyboard commands and state data, leading to failure

Cause: reference to possible fault(s) leading to failure

Severity: reference to a well-defined scale, such as "critical," "major," "minor"

Cost: cost to fix plus cost of lost potential business



Fault report template

Fault report

Location: within-system identifier, such as module or document name

Timing: phases of development during which fault was created, detected, and corrected

Symptom: type of error message reported, or activity which revealed fault (such as review)

End result: failure caused by the fault

Mechanism: how source was created, detected, corrected

Cause: type of human error that led to fault

Severity: refer to severity of resulting or potential failure

Cost: time or effort to locate and correct; can include analysis of cost had fault been identified during an earlier activity

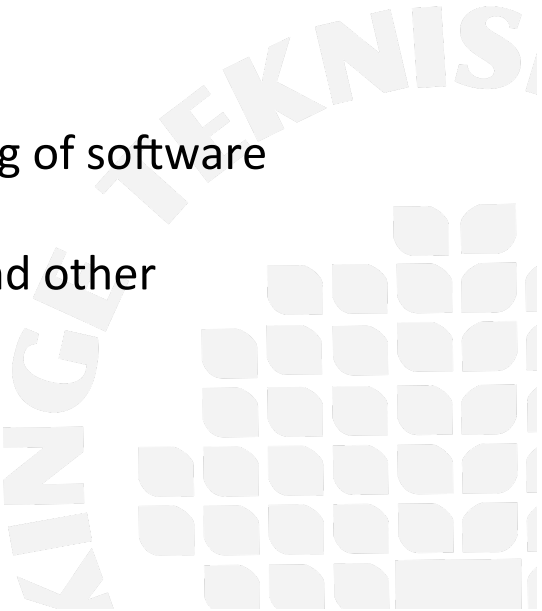
How to collect the data

- Manual data collection
  - What are the pros and cons?
- Automated data collection
  - What are the pros and cons?
- Guidelines
  - Keep procedures simple
  - Avoid unnecessary recording
  - Train staff in the need to record data and in the procedures to be used
  - Provide results of the data collection and analysis to original providers promptly and in a useful form to support their work
  - Validate

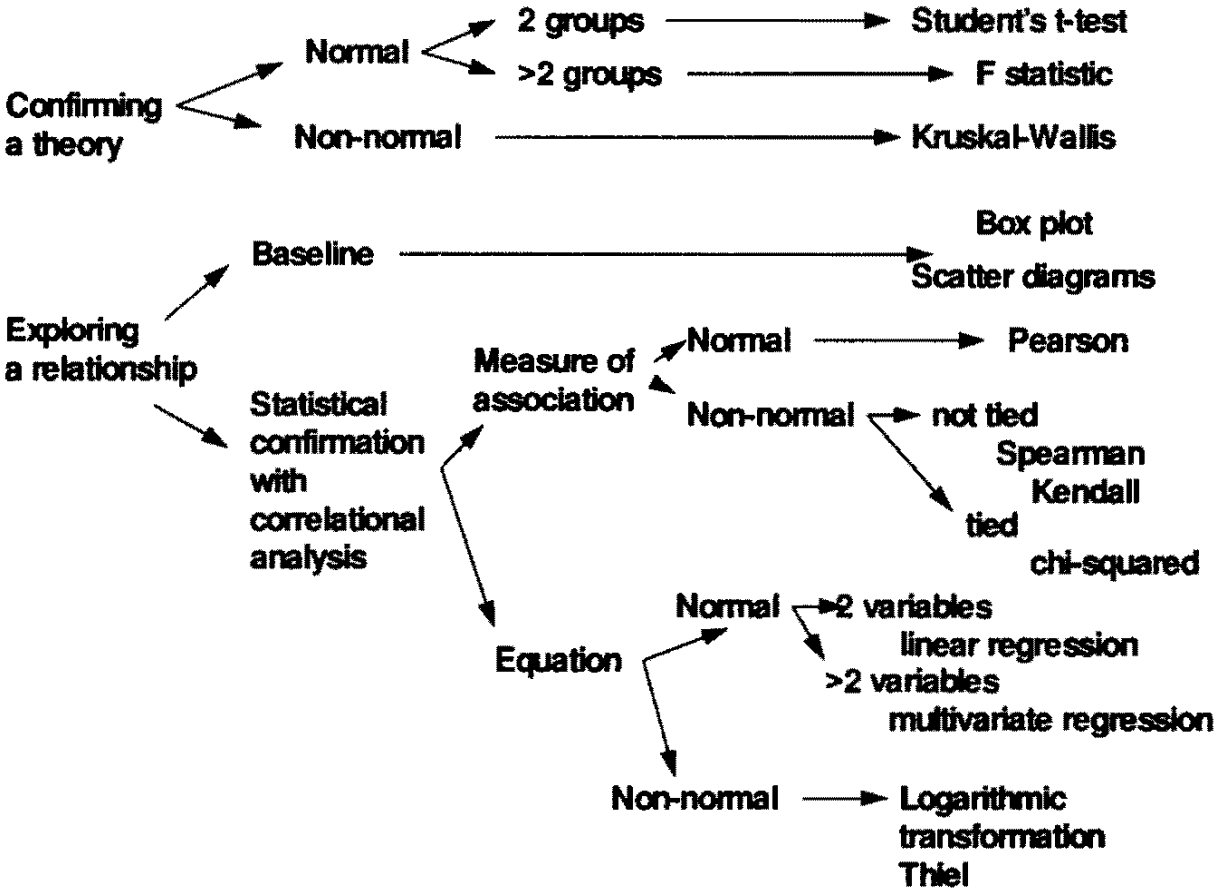
BLEKINGE INSTITUTE OF TECHNOLOGY
BTH

How to collect the data

- Data collection forms
  - E.g. data extraction forms in SLRs
  - These should be clearly defined to minimize biases/errors, and increase reliability and consistency.
- Data collection tools
  - There are many tools that support the recoding and tracking of software faults and their attributes
  - Tools that support collection and analysis of source code and other information from CVS and subversion repository logs.



Decision tree for analysis techniques



BLEKINGE INSTITUTE OF TECHNOLOGY
BTH

Acknowledgement

- Lecture notes are prepared from following sources:
  - T1: Software Metrics - A Rigorous & Practical Approach, 2nd edition, Authors: N. E. Fenton, S. L. Pfleeger, Publishers: International Thomson Computer Press, 1996, ISBN: 1-85032-275-9.