Experimentation Design in Software Engineering & Computer Science

Ways to Acquire Knowledge

Adapted from

SWE 763 : Software Engineering Experimentation

Jeff Offutt

http://www.ise.gmu.edu/~offutt/

Measuring and Science

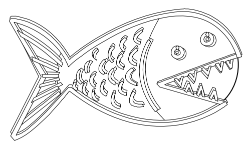
When you can measure what you are speaking about, and express it in numbers, you know something about it.

– Lord Kelvin, 1889

http://zapatopi.net/kelvin/quotes.html

What is a Scientific Test

- The Budweiser Test
  - Drinkers of another brand were given a “live” challenge – which beer is better?
  - Results?
    - 50% chose Budweiser!!!
  - Conclusion:
    - Budweiser is better !!!
- Hmmm … something’s fishy …



 3

Scientific Test

- Test: Live TV, lots of noise and confusion.
- Subjects wouldn’t be able to tell any difference, so we should expect each beer to be chosen …
- Half the time!
- There are three kinds of lies …

 4

Lies, Damn Lies, and Statistics

Berkshire Eagle, October 7, 1993 page A3 (An AP story from Boston)

Guns in the home found to increase risk of death

- People who keep guns at home nearly triple their chances of being murdered, usually by friends or relatives, but fail to protect themselves from intruders ...
- The article goes on to describe how the study was conducted, summarizes aspects of the population cross sections and conclusions of the study, and concludes with a refutation by a representative of the NRA
- Paul Blackman, research coordinator at the National Rifle Association, criticized the study ...
  - "These people were highly susceptible to homicide," he said.
  - "We know that because they were killed."

Eating and Talking

- Japanese eat very little fat and suffer fewer heart attacks than British or Americans
- On the other hand, French eat a lot of fat and also suffer fewer heart attacks than British or Americans

Conclusion: Eat what you like. It's speaking English that kills you.

Be Careful Who You Fool

The first principle is that you must not fool yourself – and you are the easiest person to fool.

– Richard Feynman (Nobel Physics, 1965)

Six Ways to Acquire Knowledge

1. Tenacity
2. Intuition
3. Authority
4. Rationalism
5. Empiricism
6. Science

1. Tenacity

Knowledge based on superstition or habit

- Examples:
  - "Good research can only be done by those in their 20s"
  - "OO design has too many subroutine calls and is too inefficient"
  - Java is too inefficient for real use
- Exposure: The more we see something, the more we like it
- Tenacity has
  - No guarantee of accuracy
  - No mechanism for error correction

2. Intuition

Guesswork: An approach that is not based on reasoning or inference

- No mechanism to separate accurate from inaccurate knowledge
- Can be valuable as a way to suggest hypotheses
- Can be very misleading

3. Authority

Accepted because it comes from a respected source

- Examples:
  - Religion
  - Totalitarian government
  - Rules our parents taught us
- No way to validate or question the knowledge
- Not the same as asking an expert – we can accept, reject, or challenge an expert
  - Teachers are experts, <u>not</u> authorities

4. Rationalism (Reasoning)

Acquisition of knowledge through reasoning

- Logical deduction
- Assume knowledge is correct if the correct reasoning process is used
- Middle ages relied almost exclusively on rationalism
- Important for theory and pure math
  - Theoretical physics … experimental physics
- Easy to reach incorrect conclusions
- Use rationalism to arrive at a hypothesis, then test with the scientific method

5. Empiricism

Acquiring knowledge through experience

- "I have experienced it, therefore it is true"
- Experience is subjective and hard to control
- "I wrote 3 programs without designing and they worked – designs are worthless!
  - Who wrote them?
  - What programs?
  - Was the the design present but just unwritten?
- Much of computer "science" is just empiricism

6. Science

Testing ideas empirically according to a specific testing procedure that is open to public inspection

- Based on reality
- Devoid of personal beliefs, perceptions, biases, attitudes, emotions
- Based on objectively observed evidence

Scientific Method

1. Identify a problem & form hypothesis
   - Problem must be clear, precise, measurable
   - Hypothesis must be testable and refutable
2. Design the experiment
   - The most creative part
3. Conduct the experiment
4. Perform hypothesis testing
   - Analyze data with appropriate statistics
5. Dissemination
   - Write legible papers and teach classes

Excellent Scientists

- Lots of <u>decent</u> scientists who are <u>excellent</u> researchers and <u>lousy</u> disseminators
- Lots of <u>decent</u> scientists who are <u>okay</u> researchers and <u>excellent</u> disseminators

Excellent scientists do both!

Be Problem Solvers

- As Computer Science & Software Engineering majors, you have proven yourselves to be good problem solvers
- Much of life is about solving problems
- Education is not about skills, it is about knowledge
- Use your education knowledge to help you:
  - Think rationally
  - Question authority
  - Solve all of life's problems

Experimentation Design in Software Engineering & Computer Science

Part II

Terminology and Concepts

Descriptive Research

Used to discover trends and tendencies

- *Observational studies*: systematic measurement of behavior
  - interrater reliability: degree to which independent observers agree on their coding of data
- *Archival studies*: examine records of past events and behaviors
- *Surveys*: asking questions about attitudes, beliefs, and behavior

Scientific Experiment

Used to understand effects

- Changes a set of variables to elicit a response
- Imposes a *treatment* on a group of *objects* or *subjects*
  - Treatment defines a way to change variables

Correlational Research

Used to establish associations between variables

- *Correlation coefficient*: statistical measure of the strength and direction of association between two variables (varies between –1.0 and +1.0)
  - *Positive correlation*: As one variable increases the other also increases
  - ***N**egative correlation*: As one variable increases the other decreases

Basic Experimental Terms

- *Hypothesis*: A testable prediction about the conditions under which an event will occur
- *Theory*: An organized set of principles used to explain observed phenomena
- *Operational Definition*: A specific way in which a variable is measured or manipulated (*treatment*)

Variables

- *Independent variable* : Manipulated by the researcher to determine if it causes a change in the dependent variable
  - Also called *factor*
- *Dependent variable* : Measured by the researcher to determine if it is affected by the IV
- *Confounding variables* : Alternative explanations for the results
- *Measured variable* : If the dependent variable cannot be directly measured, we measure a related variable to approximate

Validity

- *Internal validity* : degree to which there is certainty that the IV caused the effects on the DV
- *External validity* : degree to which the results from a study can be generalized to other situations and people
- *Conclusion validity* : degree to which conclusions relationships in the data are reasonable
- *Construct validity* : degree to which inferences can be made from the specific objects in your study to the theoretical constructs on which those objects were based

Experimental Bias

- *Bias*: A flaw in the experimental design or conduct that can change the dependent variable
  - This is often due to an inadvertent introduction of a confounding variable
- *Bias (psychology)*: A flaw introduced by an experimenter whose expectations about the outcome of the experiment can be subtly communicated to the participants in the experiment
  - Often happens when experimenters are also subjects

Example

Data flow testing finds more faults than branch testing

- Independent Variables: Data flow, branch testing
- Dependent Variable: Faults found
- Confounding Variables: tool support, characteristics of subjects, specific values chosen, knowledge of testers, …
  - Effects the internal validity
- Bias: If I invented data flow, I expect it to do better

Correlation and Causality

- *Correlated*: Two things always happen at the same time
  - Brake lights and car slowing down
- *Causality*: Understanding what causes something to happen
  - Brake light causes the car to slow down
- If A and B are correlated:
  - A causes B
  - B causes A
  - C causes A and B
  - Pressing brake activates brake light AND slows car down

Correlation and Prediction

- Correlation: if A happens, then B happens
  - Brake lights and car slowing down
- Causality: if A happens, then it causes B to happen
  - Pressing brake slows the car down
- *Predictability*: if A happens, I can predict that B will happen

We do not need to show causality to have predictability

Confusing Correlation and Causality

- In "*the old days*", we believed that being cold caused us to get colds
- Colds are caused by viruses, not temperature
- Viruses breed very well in warm, damp, low-oxygen, carbon-dioxide rich environments
- When the weather turns cold, we often close up our houses and turn up the heat … creating …
- In Virginia, we have a secondary cold season in July-August … when the weather turns hot and humid …

Cognitive Dissonance

- We feel uncomfortable when new data or a new model contradicts a previously held model
- Revising our mental model to accommodate new data is hard
  - We resist the new idea

Experimental Design

- Choosing variables, subjects, objects, process and analysis method
- *Pilot study*: small-scale experiment used to design the full experiment
  - Identify potential confounding variables
  - Refine experimental design

Avoiding Bias in Experimental Design

- *Control*: Ensuring the confounding variables do not influence the results
  - I want to measure whether maintenance programmers understand programs better by studying statecharts or reading comments
  - Comments already existed in the program, statecharts generated by experimenter
  - Statecharts were of much higher quality
  - Programmers understood statecharts better …
- Must control for differences in quality

Avoiding Errors of Judgment

- *Randomization*: Objects are assigned randomly to experimental groups
  - *Randomized block design*: Divide subjects into homogeneous blocks, then randomly assign from each block
    - Programmers: undergraduate students, MS students, PhD students, professional
- *Replication*: perform the experiment again, with different subjects, experimenters, or experiment design
  - Most reviewers will not accept replicated experiments

Experimentation Design in Software Engineering & Computer Science

Part III

Problems Specific to our Fields

Software Engineering

1. The biggest obstacle to software engineering experimentation is that our populations are unknown
   - What is a representative collection of programs?
   - Faults?
   - Developers?
2. Second : Industry won't cooperate
   - In other engineering fields, companies provide access to data, resources, processes, and people
3. Third : "Knowledge inversion" – senior scientists often do not know as much about experimentation as younger scientists

1. Unknown Populations

- How many programs are enough for external validity?
- Are seeded faults as good as natural faults?
- Does using students bias the results?
- How do we analyze our results?

1. Statistical Tests and Software

- Experimental data based on programs cannot, with validity, be subjected to inferential statistical tests since the population is unknown
- An unknown population nullifies any statistical result that would be obtained, regardless of the number of programs
- Only descriptive statistics can be used
  - For example, log linear analysis
- That is, statistical hypothesis testing, at least in the statistical sense, is not accurate

2. Industry Cooperation

- Researchers need access to data from industry to know how techniques work in practice
- Two years ago, my student applied a high-end testing technique to real-time, safety-critical software, finding several bugs
  - She was refused permission to publish, because "customers might think our software is not perfect"
- Seven years ago, a former student applied a test technique I invented to Cisco's routing software, finding several bugs, one very severe, saving millions of dollars
  - $750,000 bonus!
  - Almost fired for telling me
  - Her boss asked me to sign a non-disclosure agreement, afterwards
- Very difficult to get research funding from industry

3. Knowledge Inversion

- Every "generation" of computer scientists has taken a step forward
  - ’70s – ’80s: No validation at all
  - ’80s : We built systems
  - ’80s – ’90s : Results on small sets of data
  - ’90s : Careful experimental design, larger data sets
  - 2000s : Sophisticated statistical analysis of results
- Many journal and conference reviewers do not have the knowledge to evaluate experiments

Principles to Follow

1. Improvement is through continuous, sustained change, not technological breakthrough
   - Scientists take baby steps
   - The "big step" is the last of many
   - OO and the Web were last of thousands of baby steps
2. Take great care in your data collection
   - Identify and control variables carefully
   - Document all decisions
   - Save all data – you may have to repeat the experiment years later

Principles to Follow (2)

3. Data collection is not the goal, analysis and application are the goals
   - Don't lose the forest in the trees
   - Conclusions matter, measurement does not
4. Data are uncertain and fallible – design experiments to be fault tolerant
   - Too many variables
5. Non-developers need to collect and analyze data
   - Developers' goal is the current product, not next
   - Research lab or university who can cooperate with company

Principles to Follow (3)

6. The goal of an experiment is to help companies develop better software, cheaper
   – The goal should NOT be to publish papers
   – Sadly, tenure committees do not understand …

Summary

- SWE and CS are “inventive” fields
- Evaluation means determining how useful our inventions are
- Experimentation is the most widely used way to evaluate SWE & CS research

Expectations for evaluation increases every year