Software Quality in a Business Context

Shari Lawrence Pfleeger

WAMPS

October 2009

Overview

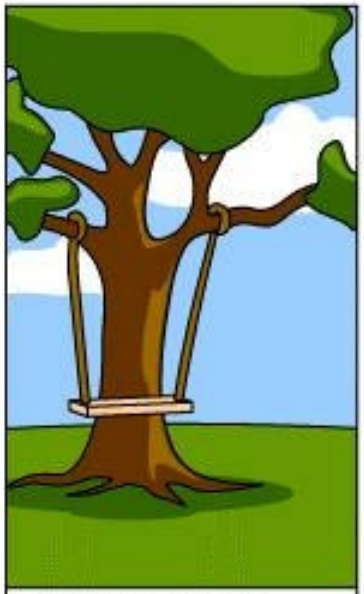
- Why measuring software quality is hard: the role of the business context.
- Three examples
  - Testing geographic profiling software
  - Usability of US-VISIT fingerprints
  - Measuring security
- Lessons learned

Perspective is All

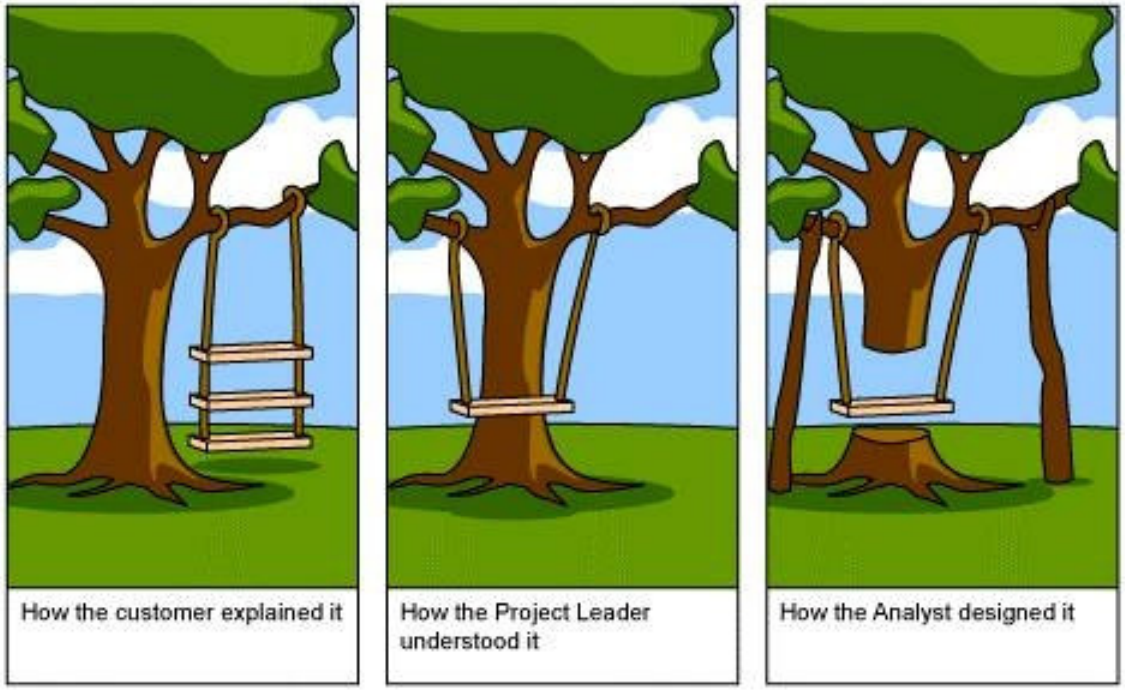
- The business goals are not usually the same as the technology goals.
- Quality depends on many things, some of which have more to do with business goals than technology goals.
- The result can be a disconnect between what is wanted and what is delivered …

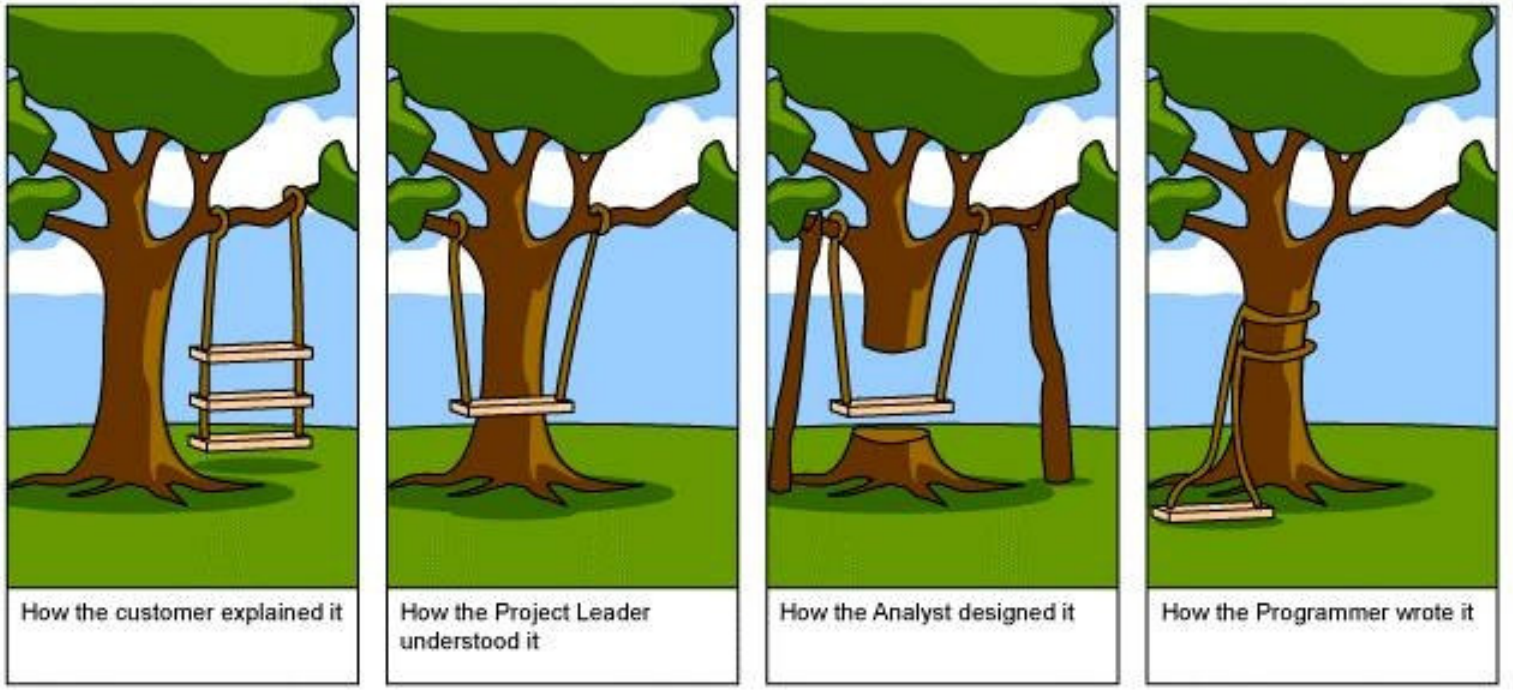
How the customer explained it

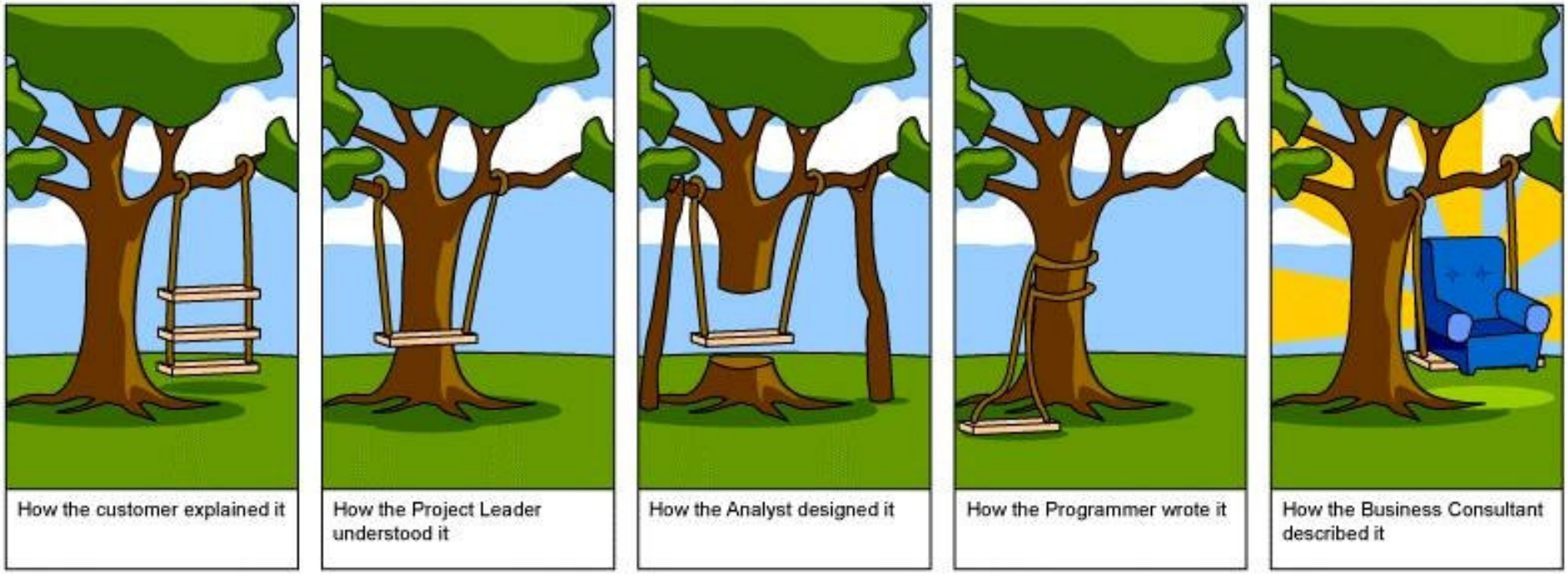
How the customer explained it



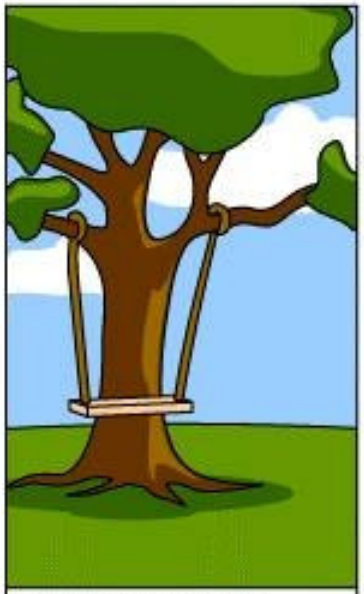
How the Project Leader understood it


How the customer explained it
How the Project Leader understood it
How the Analyst designed it


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it

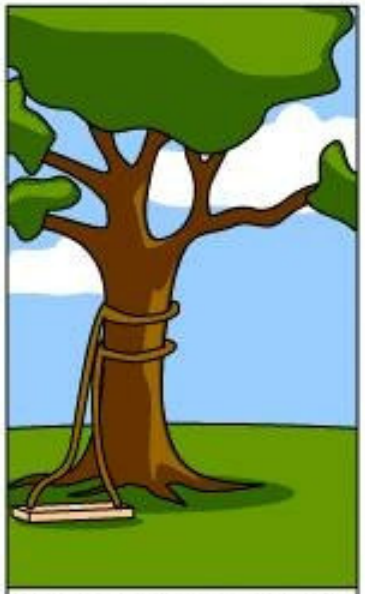

How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it

How the customer explained it



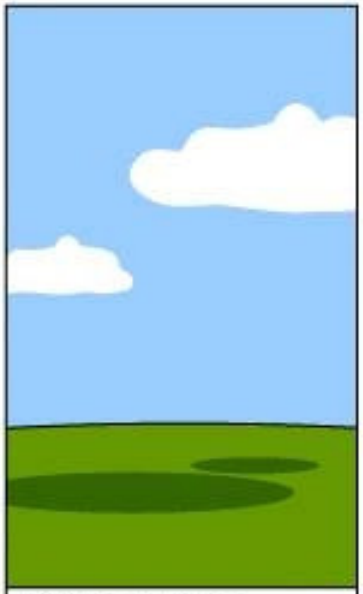
How the Project Leader understood it

How the Analyst designed it



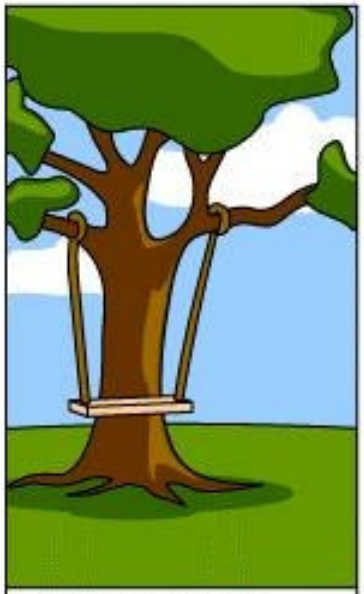
How the Programmer wrote it

How the Business Consultant described it



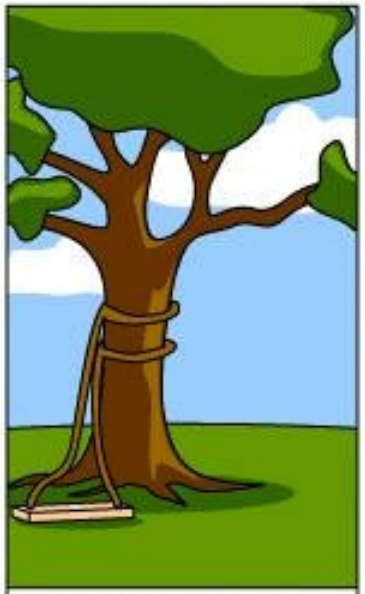
How the project was documented

How the customer explained it



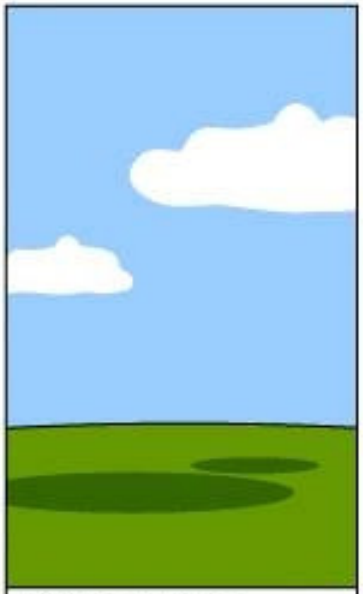
How the Project Leader understood it

How the Analyst designed it

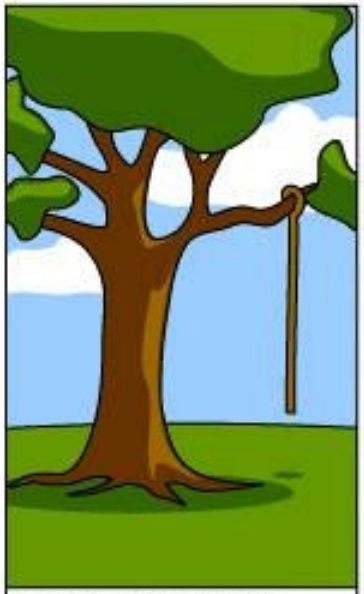


How the Programmer wrote it

How the Business Consultant described it

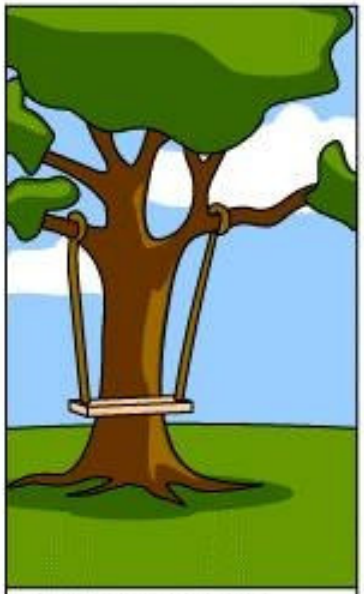


How the project was documented



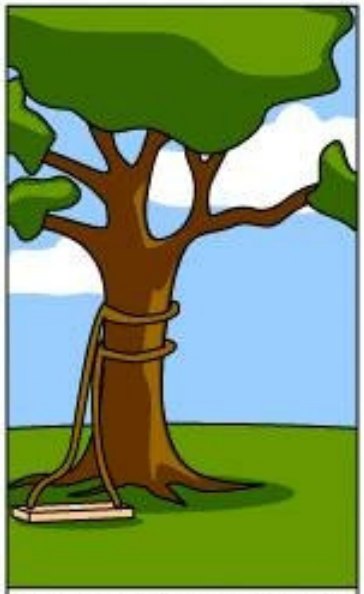
What operations installed

How the customer explained it



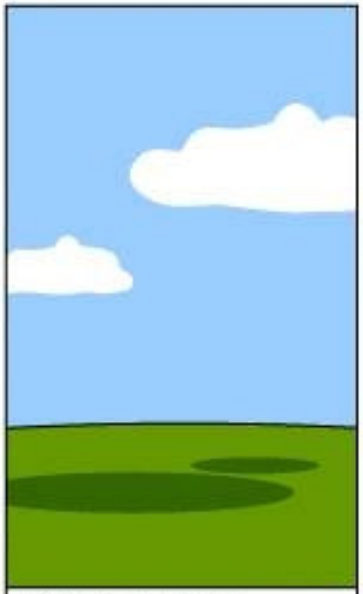
How the Project Leader understood it

How the Analyst designed it

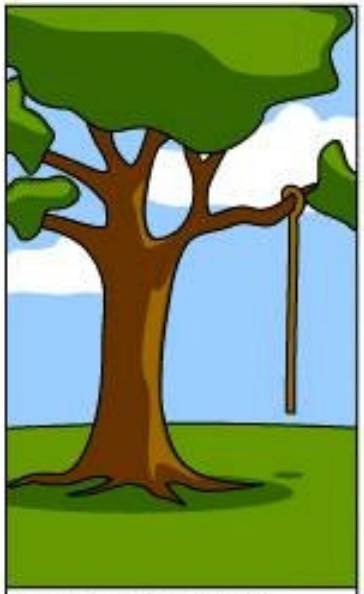


How the Programmer wrote it

How the Business Consultant described it



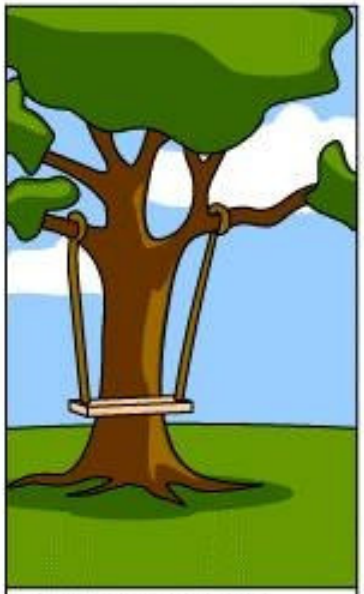
How the project was documented



What operations installed

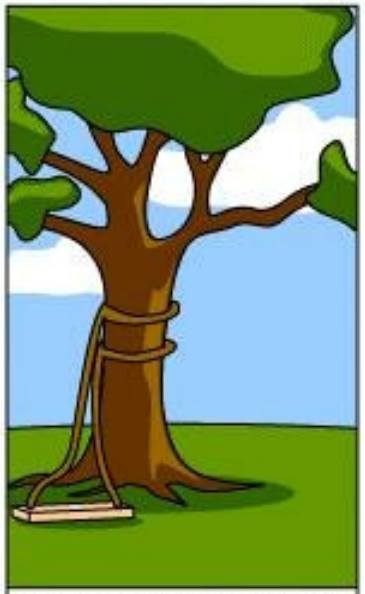
How the customer was billed

How the customer explained it



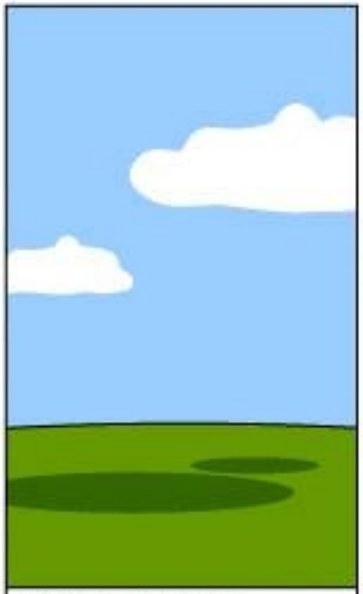
How the Project Leader understood it

How the Analyst designed it

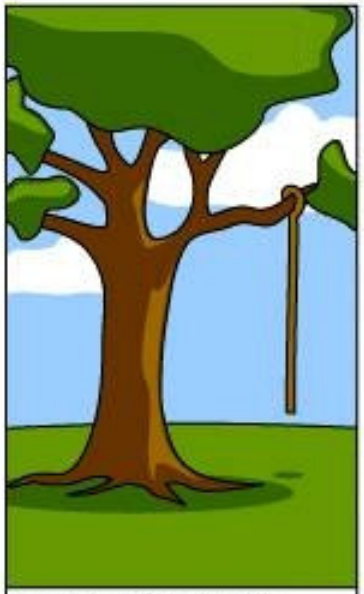


How the Programmer wrote it

How the Business Consultant described it

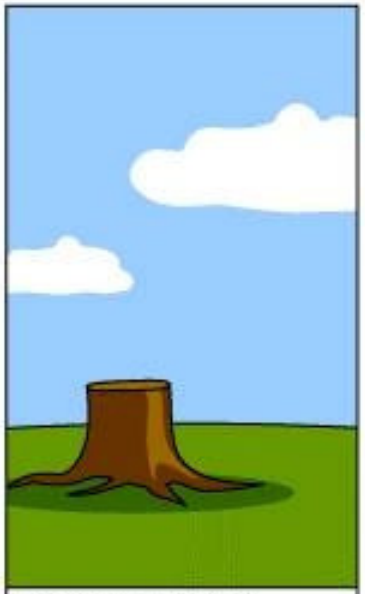


How the project was documented



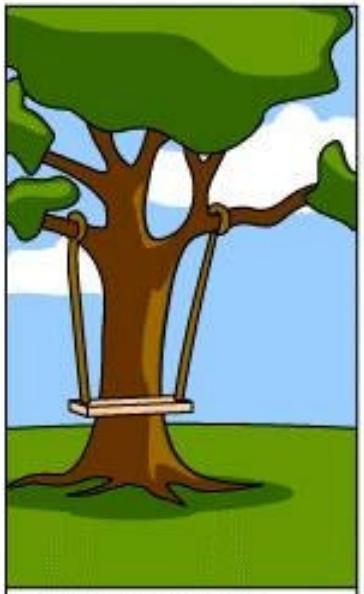
What operations installed

How the customer was billed



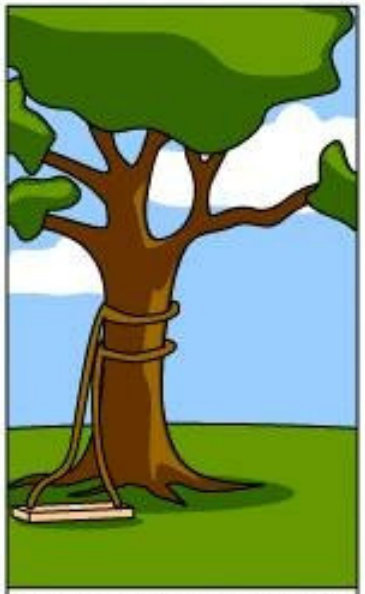
How it was supported

How the customer explained it



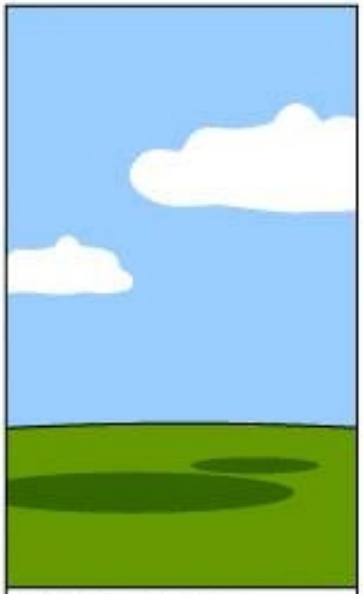
How the Project Leader understood it

How the Analyst designed it

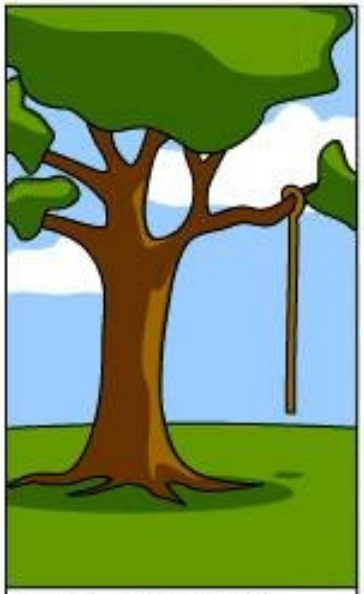


How the Programmer wrote it

How the Business Consultant described it

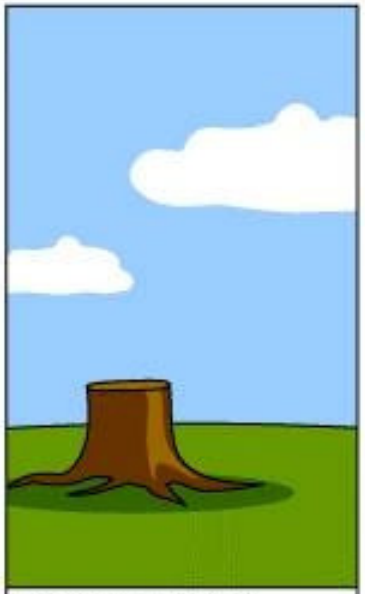


How the project was documented



What operations installed

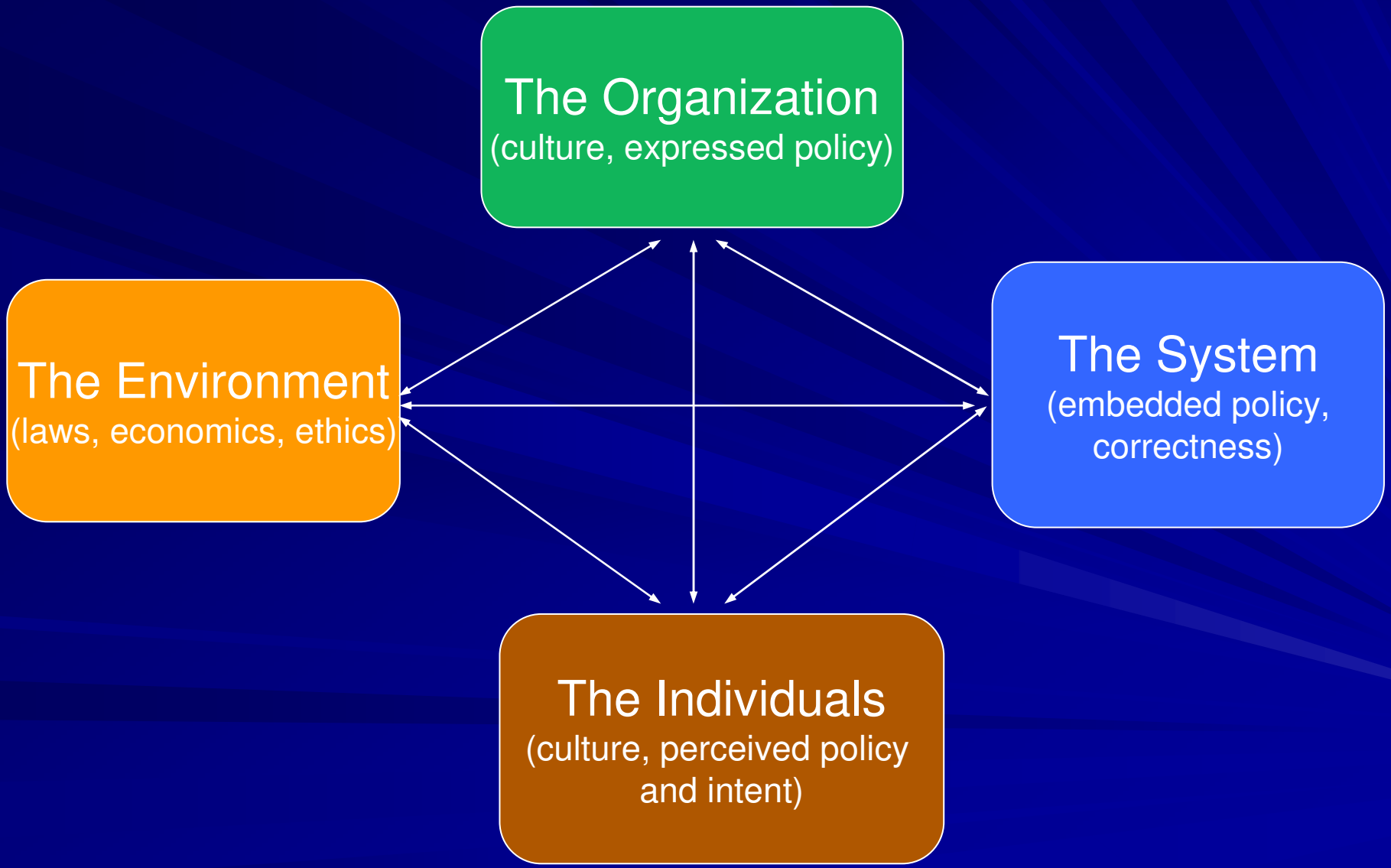
How the customer was billed



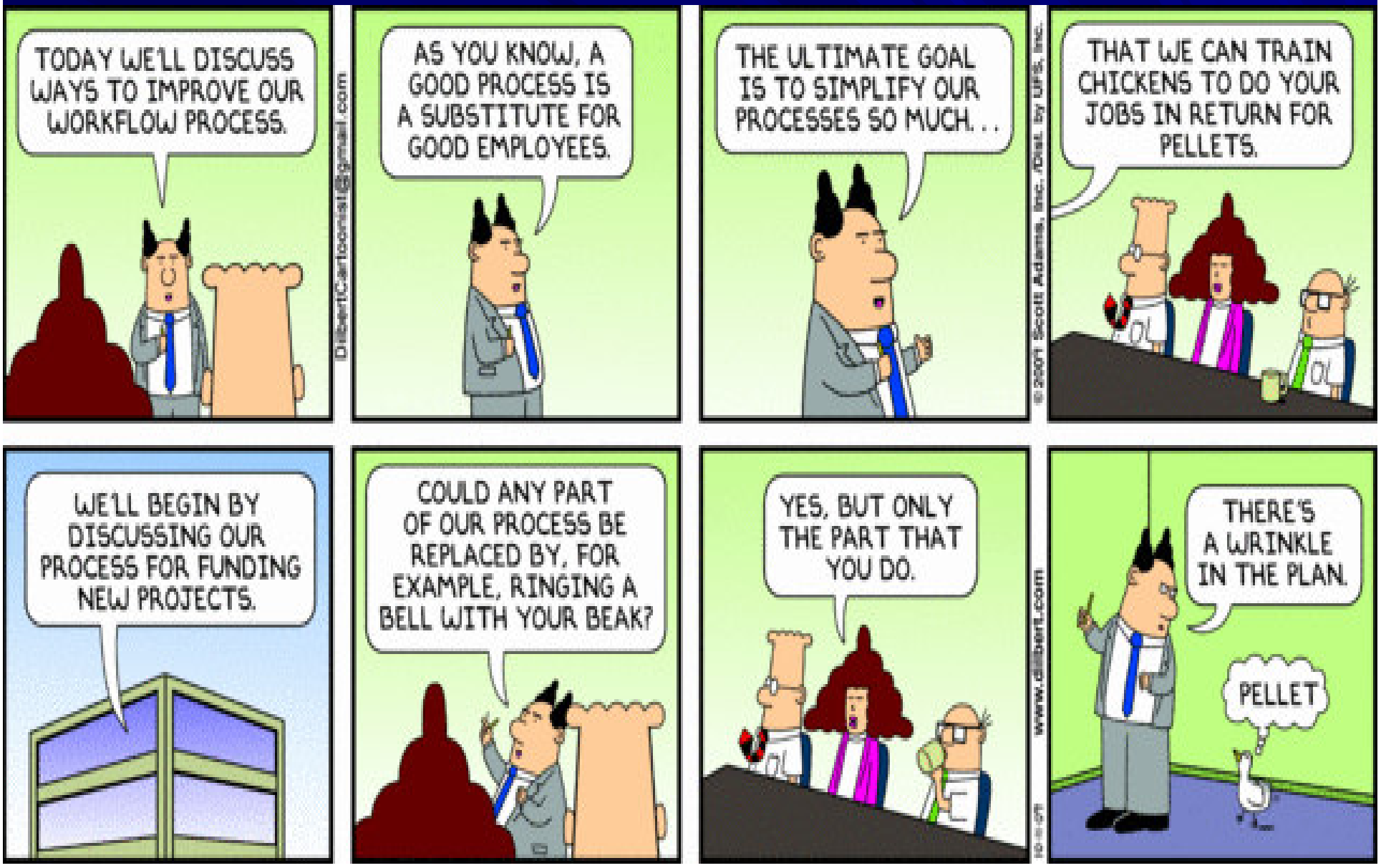
How it was supported

What the customer really needed

Four Key Perspectives on Quality



Process is Important But Not Everything

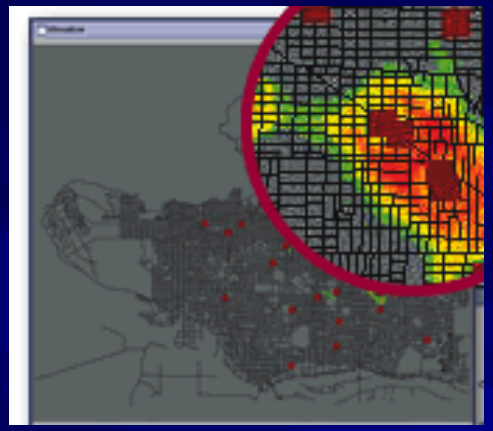


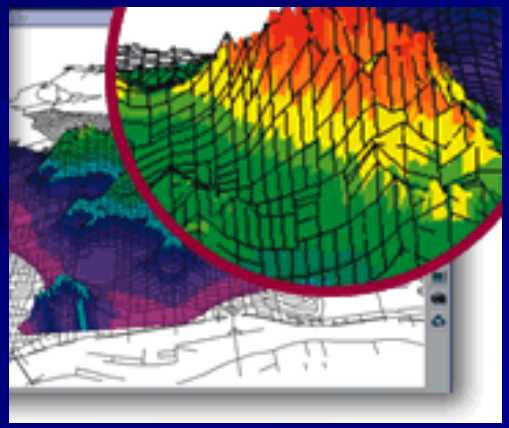
So How Do We Put the Pieces Together?

We Can Study Several Examples

- Example 1: Geographic profiling software
  - What does it do?
  - How usable is it?
  - How can we test it?
  - How does it fit in its context: law enforcement?

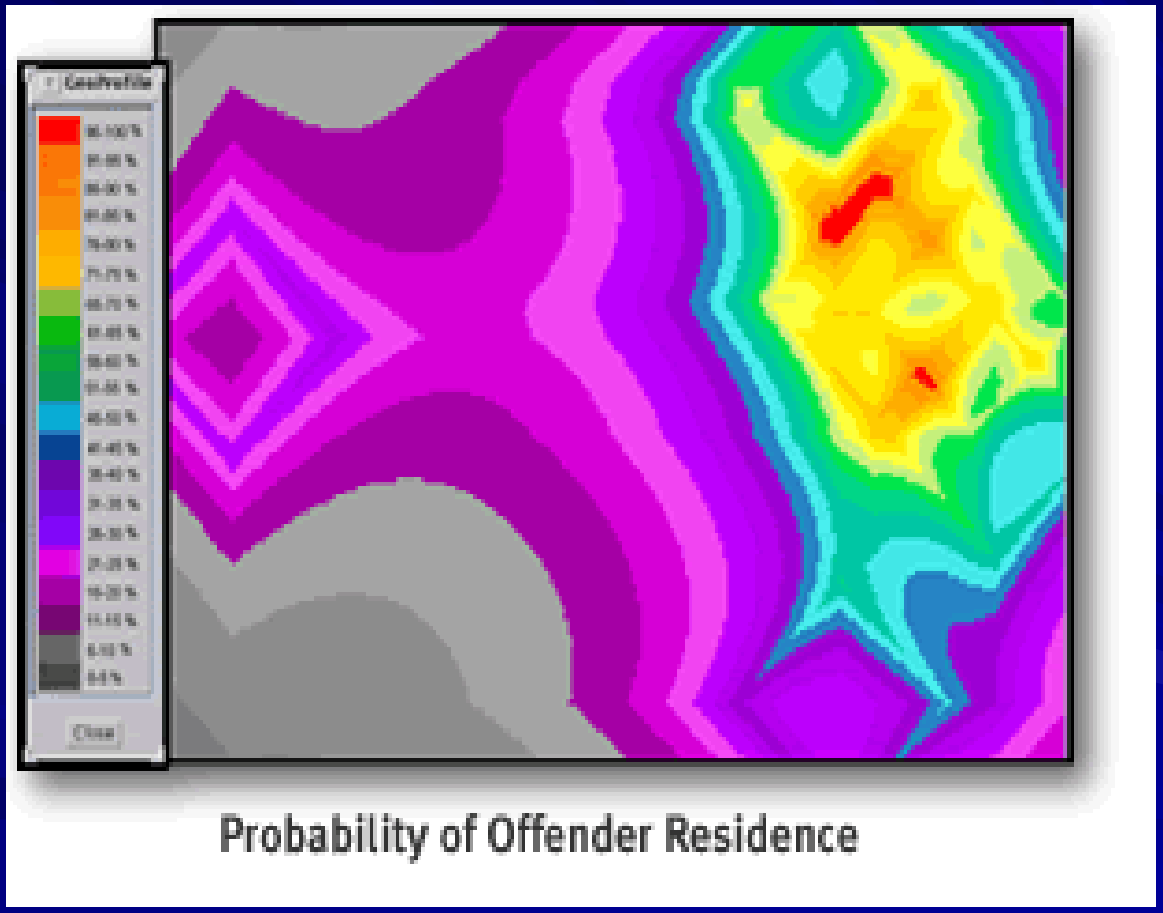
What is Geographic Profiling Software?





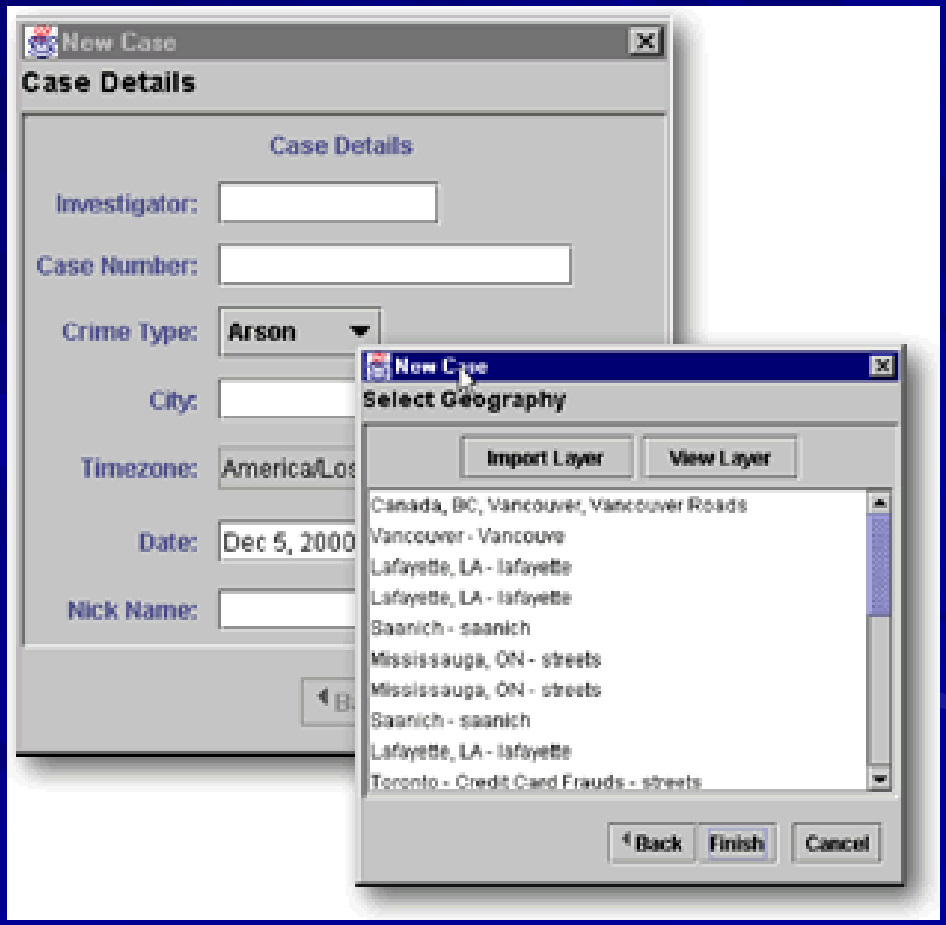
- Based on assumptions about "spatial structure" of crime patterns
- Inputs: Discrete events tied to geographic locations
- Outputs: Value maps that show most probable locations of centers of activity
- Dependencies: type of crime, criminal, geographic area
- Example: Dark red represents 70% confidence level

Example: Rigel

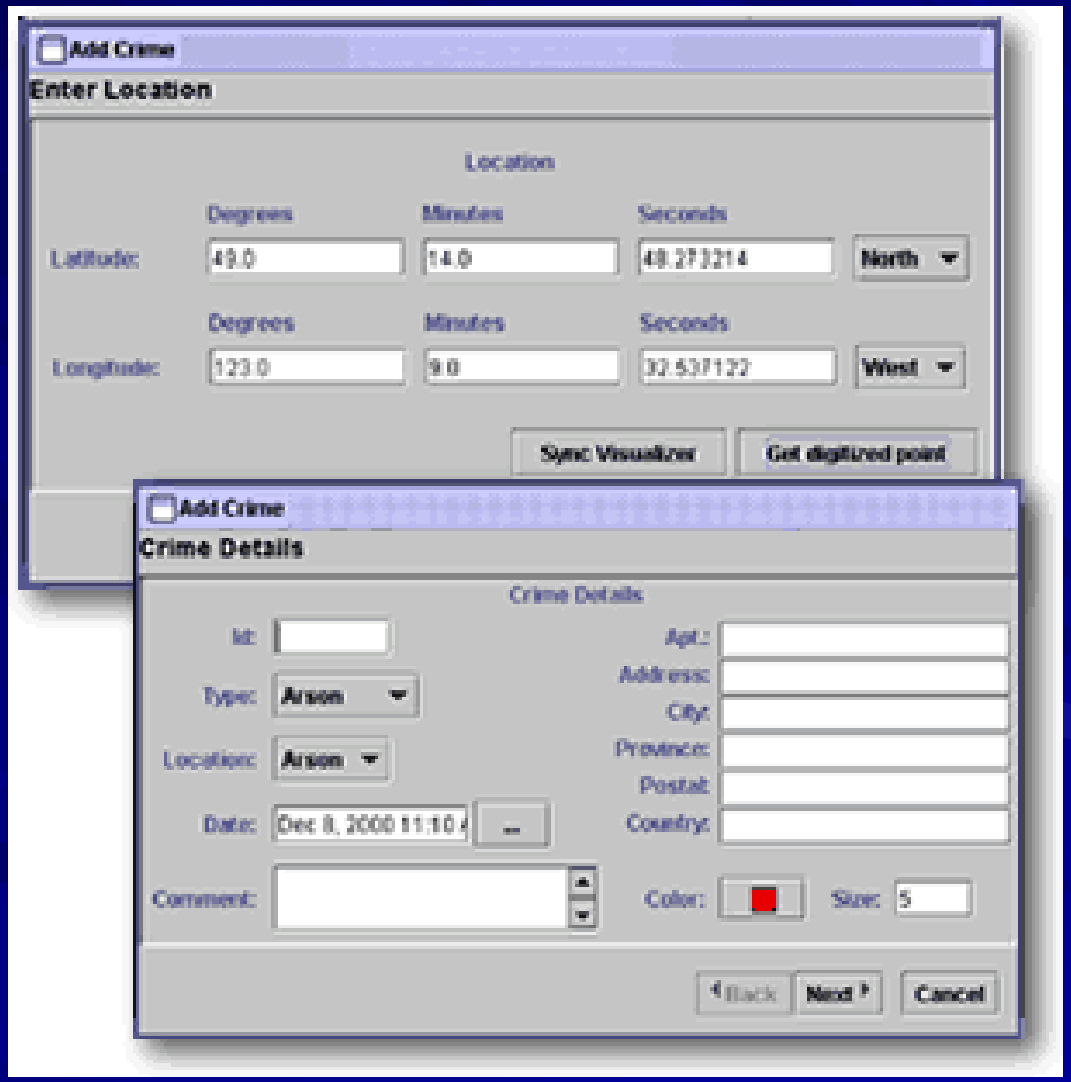


Create New Cases

- Input details for each case
  - Investigator
  - Case number
  - Crime type
  - City
  - Time zone
  - Date
  - Geography

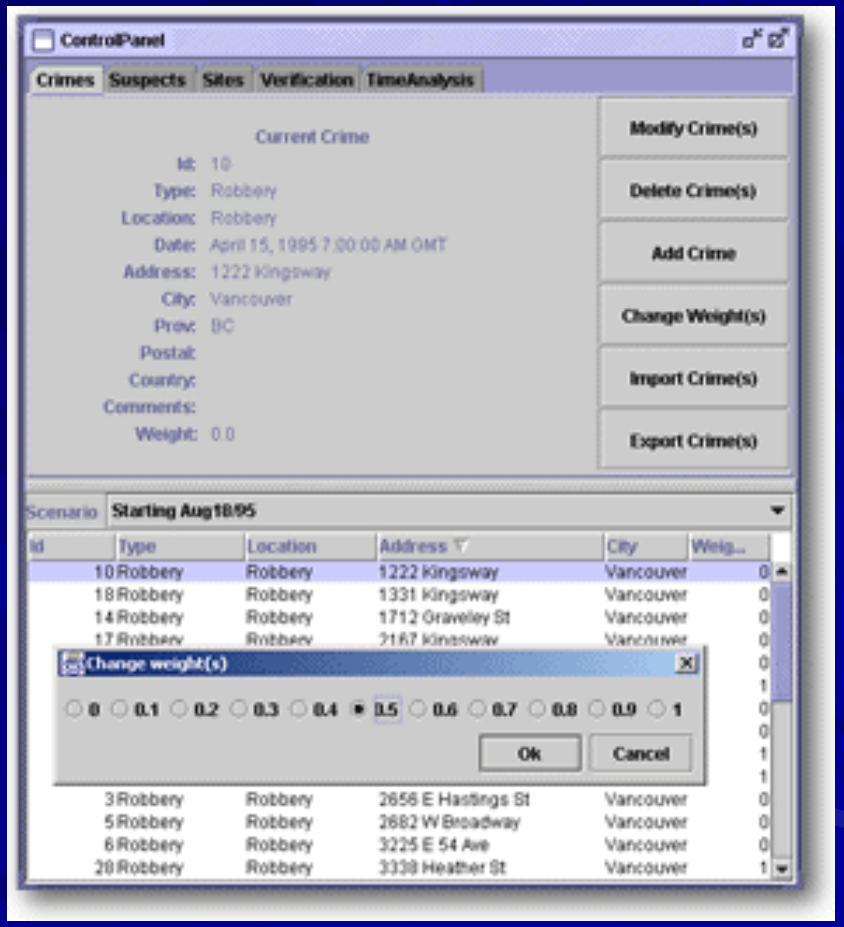


Enter Crime Site and Suspect Locations

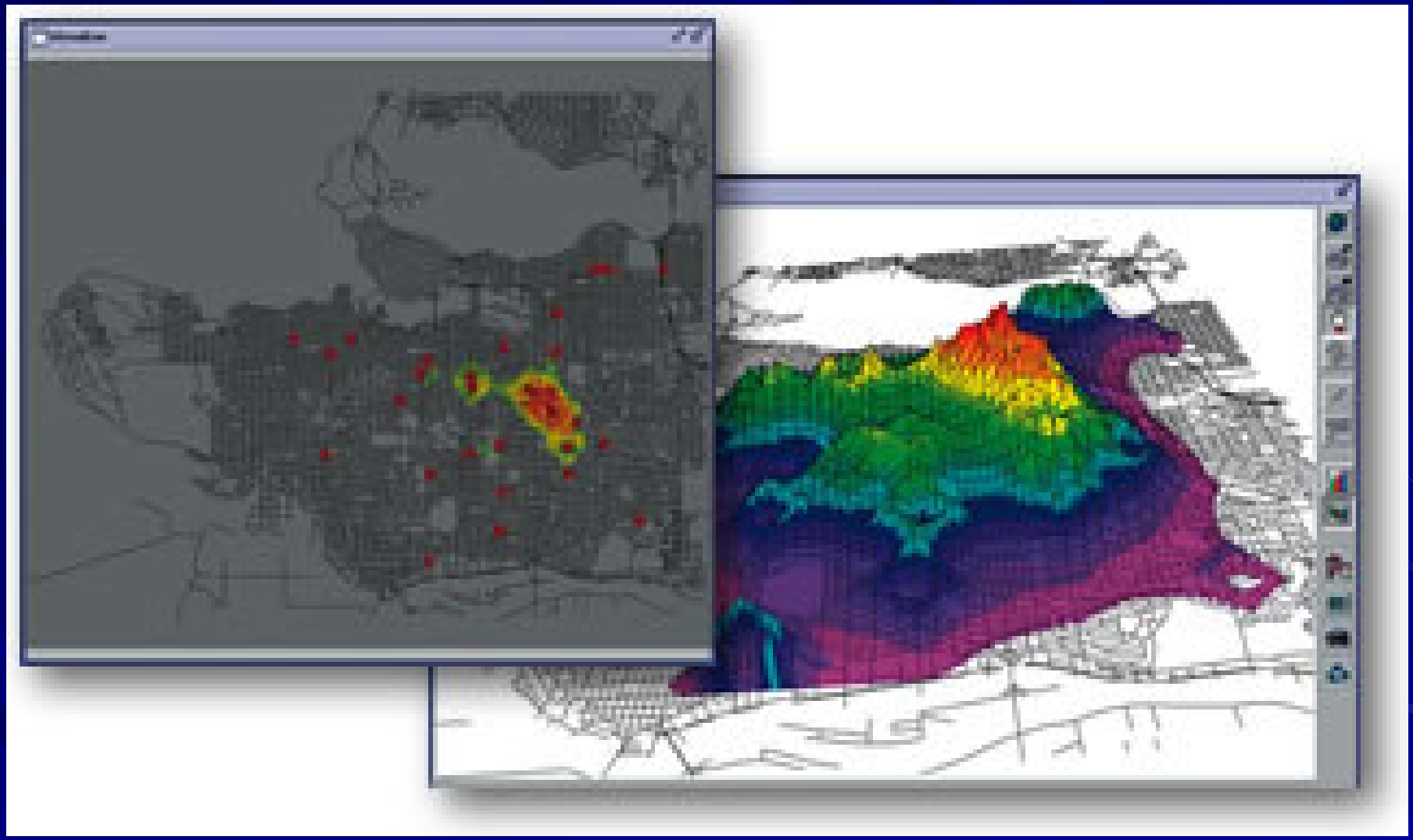


Create Crime Scenarios

- Includes weights for characteristics
- Allows importing and exporting of crime data from other databases

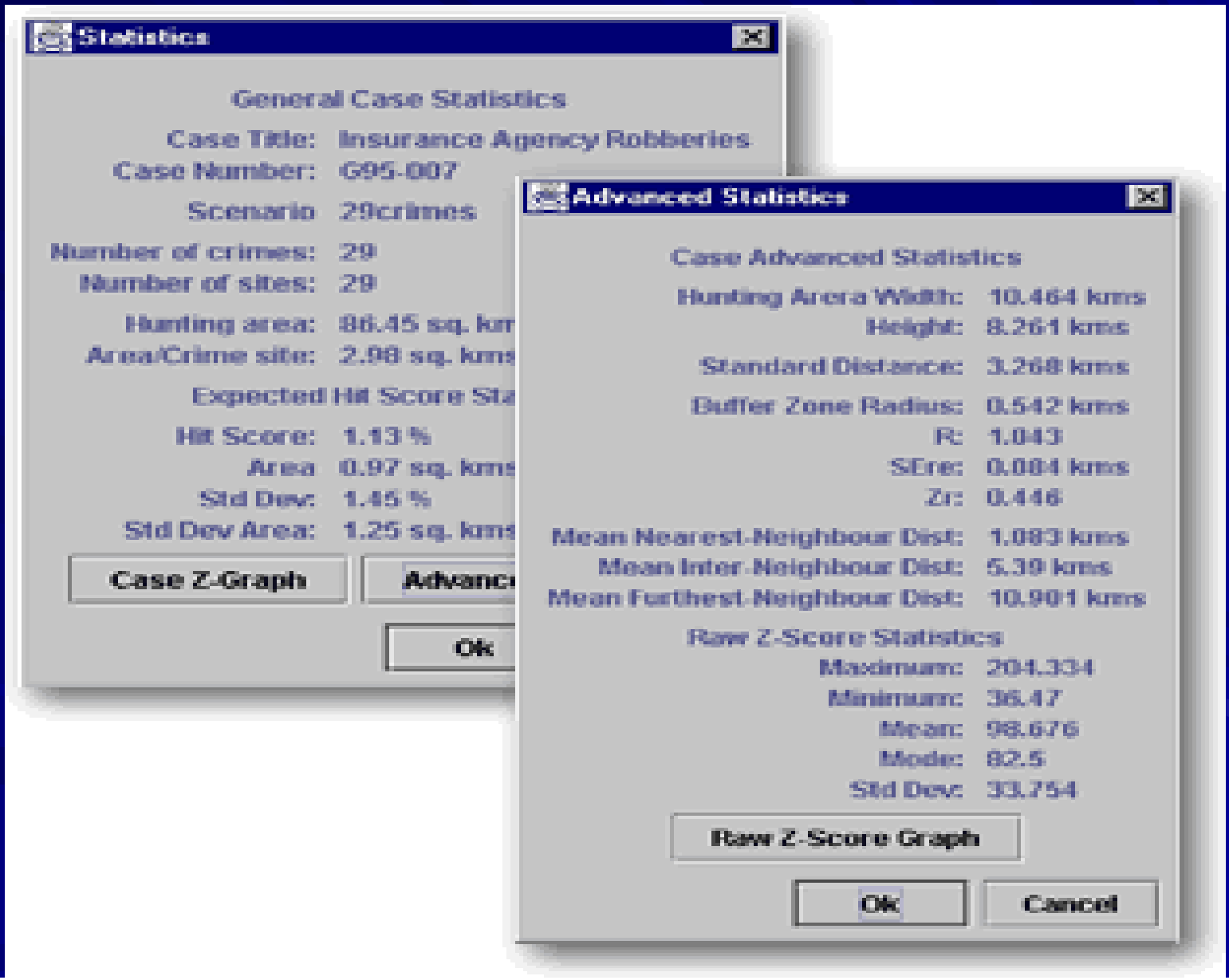


Case Analysis



Rank Suspects

Produce Statistical Reports



Questions to Ask

- How do you evaluate geographic profiling software?
- How can you assess the underlying algorithms?
- For law enforcement users, what are the best interfaces?
- How do we deal with the dearth of serial crime data?
- What about automated testing tools?

Bottom Line:
Is It Better Than Pins On a Map?

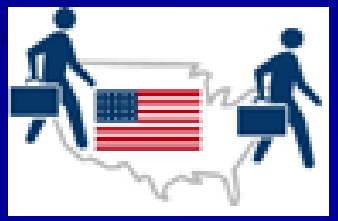
We Can Study Several Examples

- Example 2: US-VISIT software
  - What does it do?
  - How reliable are fingerprints as an identity mechanism?
  - How usable is it?
  - How can we test it?
  - How does it fit in its context: identity confirmation?

US-VISIT

Keeping America's Doors Open and Our Nation Secure

- Entry-exit program for the United States
- Goals
  - Enhance security of citizens and visitors
  - Facilitate legitimate travel and trade
  - Ensure the integrity of the immigration system
  - Protect privacy of visitors



What Does It Do?

- US-VISIT requires two things upon entry to US:
  - Ten-digit fingerprints
  - Photo
- Software compares with Watch List

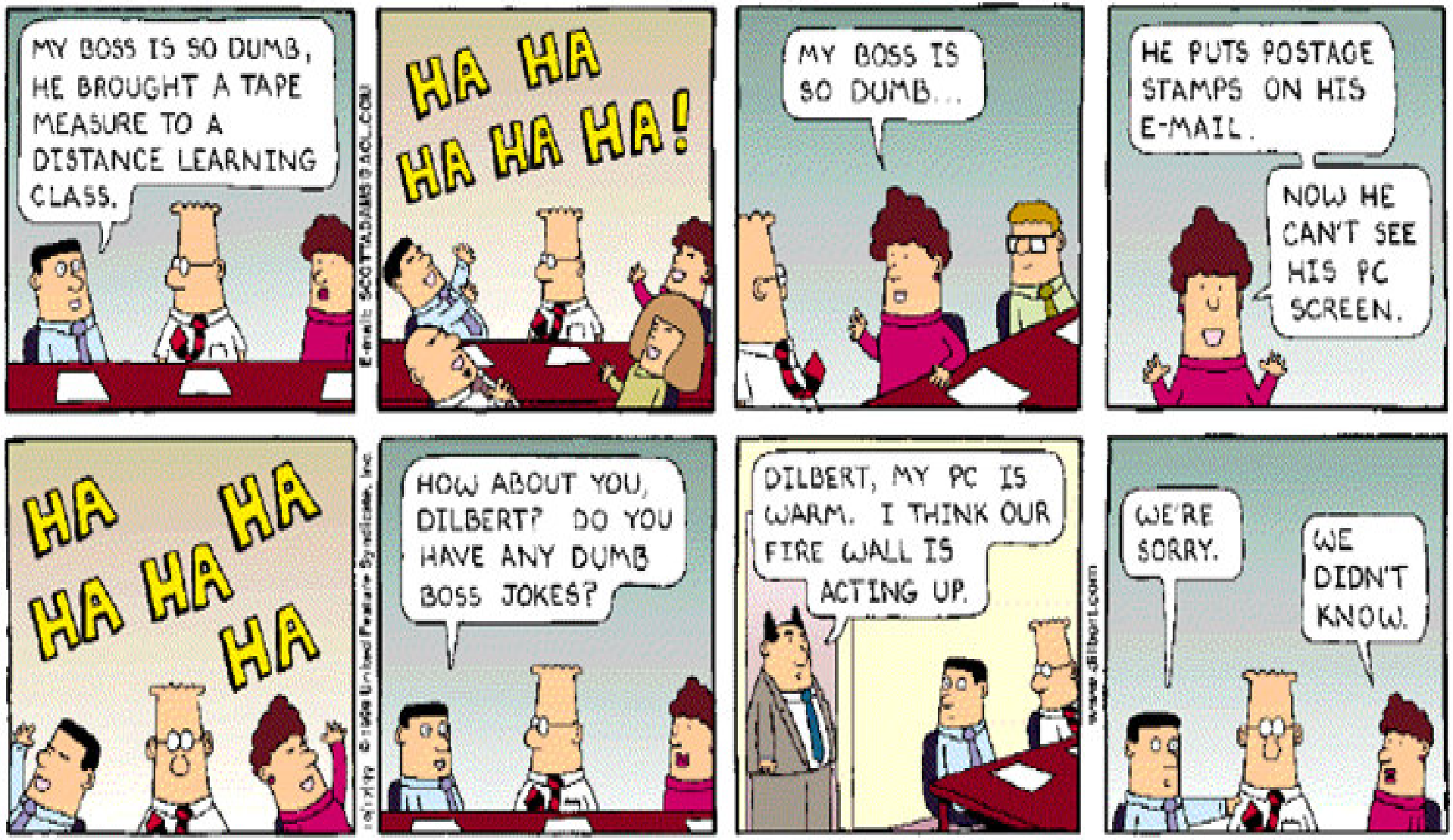
Significant Quality Issues

- Is 10-print better than 2-print?
  - Combining measures of single print quality
- False positives and false negatives
- Table height affects quality: usability issue
- Correctness of software (including conformance to policy)
- How do you test this software?

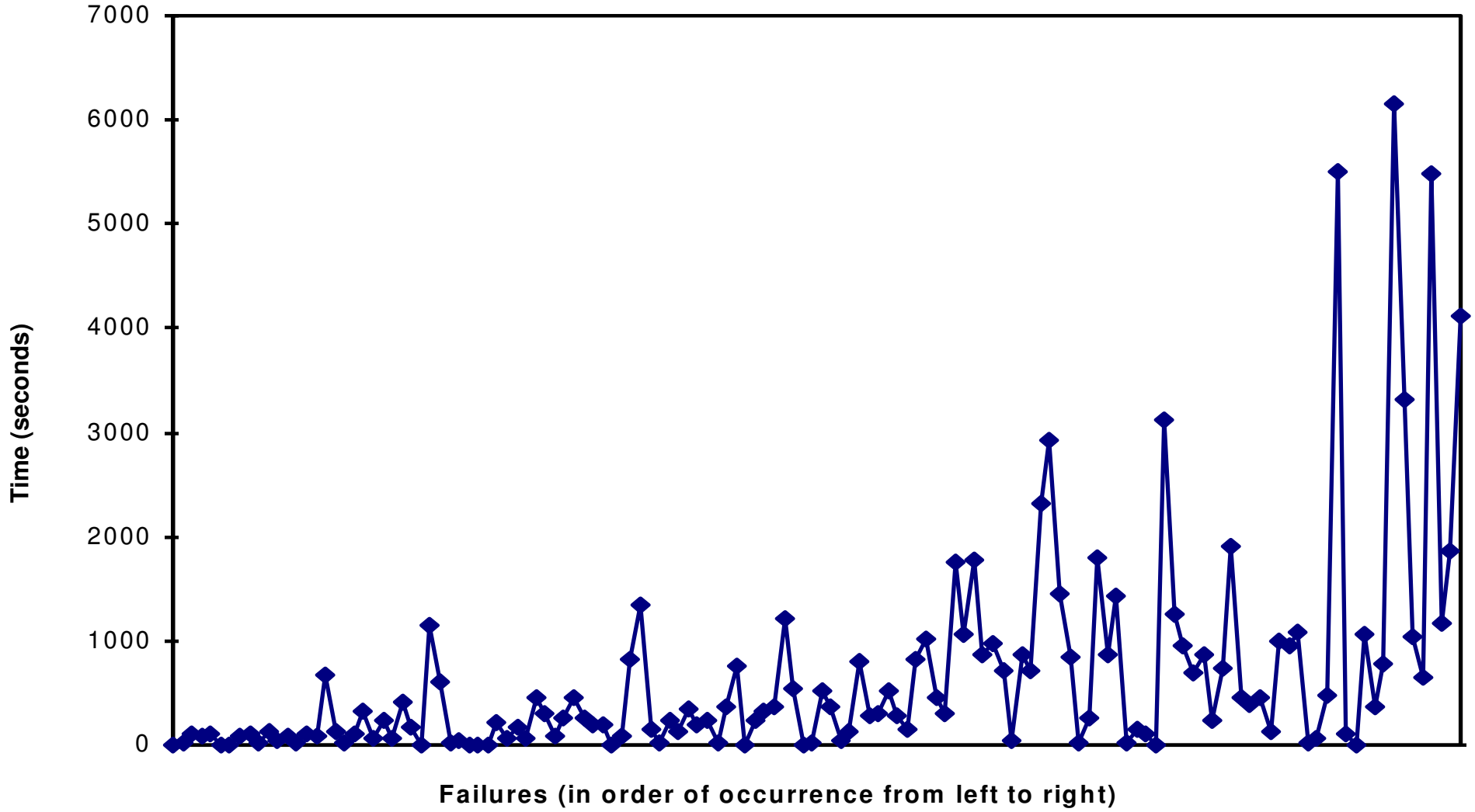
We Can Study Several Examples

- Example 3: Measuring security
  - What do we measure?
  - How credible is it?
  - How can we use it?
  - How does it fit in its context: making decisions about investment in security processes and technology?

The Wrong Way to Measure Security



Is Security Like Reliability?



Nine Reasons Why Security Measurement is Difficult

- Not all security requirements can be tested.
- Environment, abstraction and context affect it.
- Measurement and security interact.
- Few systems stand alone.
- Security is emergent and not reducible.
- The adversary changes the environment.
- Measurement is both an expectation and an organizational objective.
- We suffer from over-optimism.
- We perceive gain differently from loss.

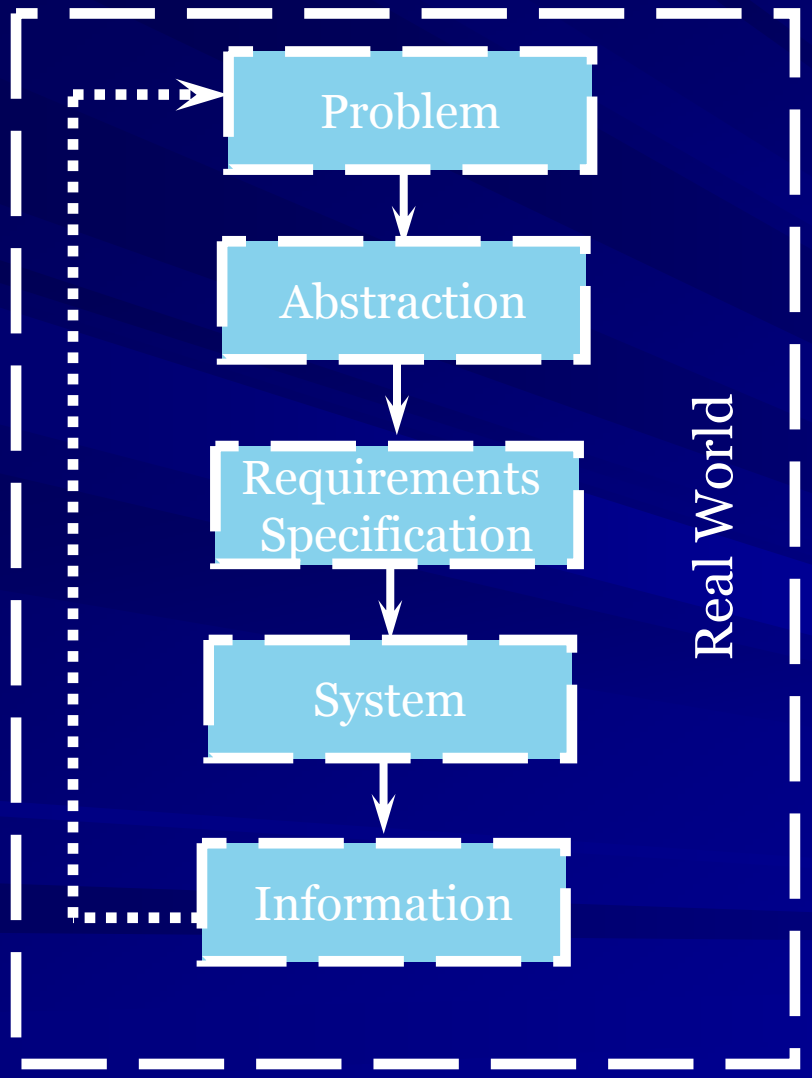
Not All Security Requirements Can Be Tested

- Charles Pfleeger: “The system will do this … and nothing more.”
- General statements about security lead to the halting problem.
- Approximations build confidence but not assurance.
- We must characterize the deployed environment and assess how well assumptions, context, threat models match operational use.
- At best: Control the factors we anticipate.

Environment, Abstraction and Context Affect Security

- Systems are evolutionary, built to incorporate change.
- Lehman: P-, S-, E-systems.
- Measuring aspects of a system's security is inherently difficult, because the system itself is inconstant.

Lehman: E-Systems



Comparison

Subject to change

Measurement and Security Interact

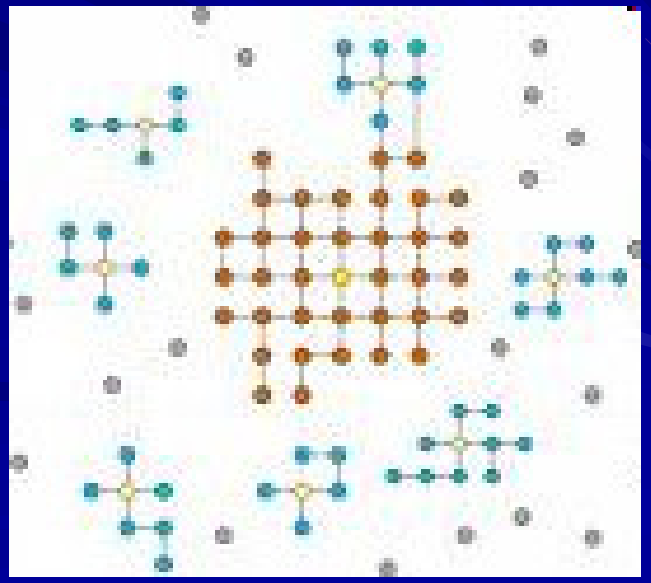
- The more detail we have about security, the more we use it to address what we think are the vulnerabilities.
- Perceived improvements in security may lead to riskier behavior à la Edward Tenner: *Why Things Bite Back*.

Few Systems Stand Alone

- System security is dependent upon partner's security.
- Trust is expressed
  - Explicitly, using mechanisms such as ACLs at network service level
  - Implicitly, for more fine-grained things
    - Computation
    - Memory use, etc.
    - As (implicit) trust in people and systems goes up, need for explicit trust goes down

Security is Emergent, Not Reducible

- Security metrics are needed across layers or models.
  - Entire system
  - Communication channel
  - Host, including application, library, and operating system software



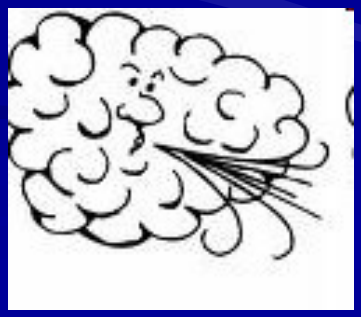
Security is Emergent, Not Reducible

- Security problems visible at higher layers may not be visible at lower ones.
  - For PCS network that controls production and distribution:
    - Confidentiality matters far less than the control's availability and integrity
  - For billing:
    - Integrity matters more
- Single security metrics that describe overall confidentiality, integrity and availability may not be as useful as measures that keep these items separate.

The Adversary Changes the Environment

The Adversary Changes the Environment

- Environment is…
  - What you're trying to do (perspective)
  - How the system is used (operational profile)
  - What you're connected to (environment)
  - What the adversary tries to do (more environment)
- This matters because the adversary invents new attack strategies that change the environment.



Measurement is an Expectation and an Organizational Objective

- Security is never absolute.
- Measurement represents an expectation of security and must be represented stochastically, with a distribution that reflects the changes wrought by the E-system and its environment, including the changing threat model.

We Suffer From Over-optimism

- Behavior and economic theory suggests that the longer we go without being attacked, the less likely we think an attack is.

We Perceive Gain Differently From Loss

- Behavior theory again.

Conclusions

- Measuring quality in theory is not so easy in practice.
  - Concepts depend on perspective
  - Perspectives hard to put together
  - Good data not available
    - Volume of data insufficient
    - Varying confidence in data quality
    - Not to mention input errors …
  - Algorithms are questionable
  - Resources are limited
  - Underlying aspects keep changing
    - Laws, rules, players, software change
  - Evaluation constrained by time available
- Expectations must be tempered by reality

So What Do We Do?

What to Do: Part 1

- Go outside traditional computer science to understand other perspectives. For example:
  - Behavior theory
  - Economics
  - Risk perception and communication
- Focus on observation to support model-building.
- Quality is a set of attributes, so use a set of metrics.

What to Do: Part 2

- Use different strokes for different folks.
  - View at ground level vs. view from 10,000 meters.
- Create a theory of relativity.
  - Prediction requires a model as well as metrics.
- Embrace uncertainty.
  - Build models that incorporate and exploit it.
  - Goal is to improve understanding, then narrow distributions.

Perguntas?