Technical Liability: Extending the Technical Debt Metaphor

Murray Cantor PhD
IBM Distinguished Engineer
mcantor@us.ibm.com
Imagine Someone offered you the following insurance deal

- We will indemnify you organization against the following future costs resulting from shipping your software
  - Data breeches
  - Data losses
  - Excess support costs (above some deductible)
  - Down time losses
  - Maybe others

- However, it will cost you $X

- What should you do?
This is not entirely a fantasy: Some examples exist

  - What is covered?
    - Lost or damaged electronic data
    - Interruption of computer operations
    - E-commerce coverage
  - What is not covered?
    - Your liability due to data loss
    - Your mistakes
    - Employee actions

- Also see [http://datainsurance.org](http://datainsurance.org)
Suppose further, some accountant says:

- “Oh, I see this is like any other kind of insurance. Either we buy it or we self-insure. Either way this code creates an liability”

- This liability must be accounted for on the books (maybe)

- In any case, the fair price of the self-insurance affects the asset value of the code.
Questions arise

- How to calculate $X$?
- Is $X$ like driving insurance, good drivers get better rates?
  - What can you do the code to lower $X$?
- Should we buy the insurance or self insure?
So how to compute X: Some observations

- **Context dependent**
  - More liability assumed for next release of an avionics dashboard than the next release of angry birds
  - Explore the range of future costs

- **The liability involves the likelihood of future events:**
  - X has to be approached probabilistically, in particular it requires predictive analytics
  - X depends on the lifetime of the code or at least
  - X is measured in currency
Uncertain values are captured as ‘Random Variables’ characterized by a probability distribution functions.
How to find the distributions

- **Initial**
  - Build some sort of model of the future with random variables
  - Apply Monte Carlo simulation to get initial distribution

- **Refine**
  - Gather actuals as they occur
  - Use Bayesian techniques to refine estimates using actuals as evidence
Some approaches to predictive analytics

- **Parametric models, curve fitting**
  - Assume some *a priori* function of project duration based on program parameters (klocs, staff members, complexity, nature of program, ...)
  - Fit multivariate curves to actuals over project population
  - Examples
    - Exponential curves: COCOMO\(^1\), SEER\(^2\)
    - Rayleigh Distribution: SLIM\(^3\)
  - Note: dispersion is captured as cone of uncertainty

- **Decision trees\(^4\)**
  - Model development as a set of decision points
  - Each decision point is specified as a discrete probability distribution.
  - Outcome distribution is found applying Monte Carlo simulation
  - Related to Real Options, Black-Scholes

- **Expert opinion\(^5\)**
  - Ask experts for best case, worse case, likely case
  - Train and calibrate organization using feedback from actuals
  - Convert experts opinion to triangular distributions

---

\(^1\)http://csse.usc.edu/csse/research/COCOMOII/cocomo_main.html
\(^2\)http://galorath.com/?gclid=CKChxejH9bgCFYai4AodhuAAHw
\(^3\)http://www.qsm.com/tools/slim-estimate?gclid=CL3RxJH9bgCFQai4AodrTwA9R
Monte Carlo simulation helps us find the distributions

- Monte Carlo simulation allows us to understand the possible outcomes of a process, and their likelihoods.

- For example, suppose service costs for
  - 2014 is estimated: Best case $200K, Likely $300K and Worse Case $600K
  - 2015 is estimated: Best case $300K, Likely $450K and Worse Case $800K
  - ...

- To find the distribution of the sum, convert informed guesses into triangular distributions.

- Thousands of times sample the distributions and record the sums.

- Normalize histogram of the sums to get distribution of outcomes.
Techniques for Learning From Actuals

1. **Leverage actual distribution**
   - The actuals form a distribution
   - We can sample from this distribution during Monte Carlo simulation

2. **Predicting based on relationships between estimates and actuals (regression)**
   - If estimates and actuals are strongly correlated, regression analysis can be useful in predicting actuals from estimates

3. **Predicting based on learned relationships between multiple variables and actuals**

4. **Bayesian Networks**
   - After 50 tasks complete, we have new evidence:
     - DB tasks: effort = [11d, 16d, 42d], type = 20%
     - Analytics tasks: effort = [4d, 16d, 20d], type = 75%
     - Mobile tasks: effort = [1d, 2d, 4d], type = 5%
   - We can update the conditional probabilities to reflect this learned evidence
Technical liability is the cost of insuring against all of the future costs that may result over the lifetime of the code.
Technical Liability Extends the Technical Debt Metaphor: It is cost assumed by delivering code

**Technical Debt**

- Deterministic (mostly)
- Based on what is known
- Relatively simple computations

**Technical Liability**

- Probabilistic
- Includes management of uncertainty
- Requires more advanced predictive analytics
Challenges

- Unlike an insurance company, an IT shop may not have big population across which to distribute the risks
- People on their own are not very good at assessing risk.
  - We have biases
  - Hence the rise of predictive decision support tools
- Each kind of liability has its own predictive model: Support, security, data integrity, ...
- These models involve skills not common in our community
Discussion: Next Steps

- Agree on and characterize the problem
- Agree on the taxonomy, flavors of liability
- Collaborate with subject matter experts to build out models for each flavor of liability (there is a lot of expertise out there)
  - Total support costs including executives getting on airplanes
  - Product liabilities, recalls, …
  - Security
  - …
- …
“The only way to predict the future is to have power to shape the future.”

~ Eric Hoffer

Questions
Thank You.