The CIO and lead architects are discussing next year’s budget. They discuss the impact of quality.

- They need to prepare to defend their budget to the CFO.
- They have to balance their limited budget between investments in quality or functionality.
- They want to systematically reason about when investing in quality is the better choice:
  - Demand for features may at first seem more urgent.
They come to some agreement on tackling the problem:

- Quality matters because it affects the bottom line of the business
  - Impacts of poor quality can be very expensive

- On a approach: Measure the business value of the software with or without quality improvements
The product Value (NPV) consists of business case elements

For a product, P:

\[ NPV(P) = NPV(\text{Benefits}) - NPV(\text{Dev Costs}) - NPV(\text{After Delivery Costs}) \]

- **Benefits**: How you measure success: revenue, cost avoidance, new customers, labor savings, mission

- **Development costs**: coding, deployment, distribution, ...

- **After market costs**:
  - Service, warrantee
  - Technical debt: cost of addressing code shortfalls
  - Technical liability: cost of addressing bad business outcomes

Can be most impactful
Assuming technical liability is like self-insuring your software.

Imagine some agent offered you the following insurance deal:

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

- “However, it will cost you $X”

What would you think?
The question arises, “What is the fair price of the policy?”

In particular:

- How to calculate the liability?
- What should I do to the code to lower the liability?
Computing technical liability

- Context dependent – More liability assumed for next release of an avionics dashboard than the next release of angry birds

- Uncertain – as The liability involves the likelihood of future events:
  - X has to be approached probabilistically, in particular it requires predictive analytics
  - The liability is best described by a probability distribution

- X is monetary – it is measured in currency

A fair value of X might be the mean of this distribution
For software, costs and benefits come in the future.

Hard fact of life: **Software is not worth what you have spent on it**

\[ NPV = \sum_{i=0}^{n} \frac{FV_i}{(1 + r)^i} \]

- \( NPV \) = Net present value
- \( FV_i \) = Future Value at period \( i \)
- \( r \) = discount rate

NPV is how finance evaluates worth now of monies earned or spent in the future. For example how does one today value a flow of future revenues given an inflation rate.
Unlike investing in a bond, the future costs and benefits are uncertain

For software, IT projects

- The future values in NPV equation are uncertain – they must be described probabilistically, as a distribution

- Hence the NPV, RoI are also distributions
The inputs capture beliefs about the future values

- Capture expert opinion of range of values
  - Best case - Zero probability of FV less than ($100K),
  - Worse case - Zero probability of FV greater than worse case ($170K),
  - Most likely case - Most probable case ($120K)

- Like planning poker

- Capture as triangular distributions for prediction tools
The output captures the probabilities of the NPV

- For example, for a product the NPV might be a common bell-shaped distribution with expected value of $100K and standard deviation of $30.

- Based on the input, there is a
  - 10% chance the NPV will be less than $61K
  - 10% chance it could be more than $138K

- Whether to invest depends on your risk appetite.

Calculated with Monte Carlo simulation.
For portfolio management

To get a portfolio view, plot value, risk of projects.
A Portfolio View
A Portfolio View

Portfolio

Low Risk, High Reward
Low Risk, Low Reward
High Risk, High Reward
High Risk, Low Reward
Quality affects NPV

For a product, P:
\[ \text{NPV}(P) = \text{NPV(Benefits)} - \text{NPV(Dev Costs)} - \text{NPV(After Delivery Costs)} \]

Investing in quality *may* affect

- **Benefits**
  - Higher because of better acceptance, utility
  - Lower because of delayed market release or deployment

- **Dev Costs**
  - Probably higher, drives need for efficiency

- **After delivery costs**
  - Likely lower

*Don’t know the RoI without a model*
The quality investment decision

At some point, you get diminishing returns on quality, the money may be spent elsewhere.

In fact, $RoI$ can turn negative. When it does, you know it is time to ship.

Also lowering the costs of enhancing quality enables shipping more higher quality with positive ROI.

$$RoI = \frac{NPV(P_1) - NPV(P_0)}{cost(P_1) - cost(P_0)}$$

$P_1 =$ product with enhanced quality

$P_0 =$ product with initial quality
It is time

- Economic pressures are growing

- Techniques and tools are readily available
  - Bayesian Net
  - Monte Carlo
  - Open Source R packages

- Knowledge is
  - Online courses
Why Bother?

- No longer flying blind: “To measure is to know.”

- With these techniques, software and IT become
  - a more mature business function
  - a value center (not just a cost center)

- Enable well-informed, information-driven conversations
  - With stakeholders, e.g. CFO
  - With staff (avoid the loudest wins value conversation)
  - Have better informed budget discussions

- Promotes integration of concerns among subject matter experts
To start

- These models involve skills not common, but should be, in our community – start building the capabilities:
  - Business case
  - Analytics
  - Technical debt
  - Liabilities experts

- Start with simple models
  - They just have to be good enough to support decision.
  - Add detail only if need, avoid complexity

*Perfect is the enemy of the good*
To build the models:

- Start with, or create the business case!
  - 3 NPV terms: Benefits, dev costs, after delivery costs.
  - Usually involves different SME’s stakeholders
    - Benefits: Marketing, business analysts, …
    - Dev Costs: Developments
    - After market: support, liabilities experts, technical debts owners
  - Each NPV will have own own model
    - E.g. revenue (benefit) may depend on estimate of price, market size, and penetration

- Elicit uncertainties
  - Best case, worse case, expected case for inputs to the models

- Build integrated model in appropriate tools
“The only way to predict the future is to have power to shape the future.”

~ Eric Hoffer

Thank You