What is a Risk?
by Linda Westfall

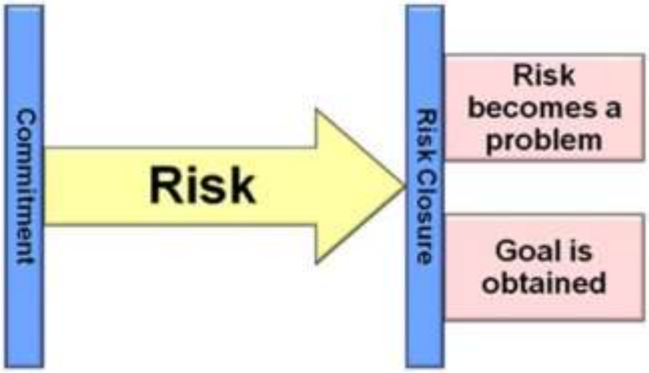
In our global economy, the possibilities of reward are high, but so is the potential for disaster. Risks exist whether we acknowledge them or not. Sticking our heads in the sand and ignoring the risks can lead to "unpleasant surprises" when some of those risks turn into actual problems. Tom Gilb's risk principle illustrates the need for enterprise-wide risk management. "If you don't actively attack the risks, they will actively attack you." To successfully manage our projects and products and reap the rewards, we must learn to identify, analyze, plan for, track, and control our risks.

What is a Risk?

That said, what do we mean when we are talking about a "risk"? A risk is simply the possibility of a problem occurring sometime in the future. If it is already a problem, put it on the problem list and deal with it -- it is not a risk. If it has a 100% chance of happening, it is already a problem – a risk has some possibility of not turning into a problem.

There are many risks in creating high-quality software-intensive products on time and within budget. Developing software is risky with ever-increasing software complexity, increasing demands for better software with more functionality, and decreasing time to market windows. When teams don't manage these risks, they leave their projects vulnerable to factors that can cause significant rework, cost or schedule over-runs, and delivered software that doesn't match the intended use. For example, software having safety, security, usability, or functionality gaps or other software project failures.

When Does a Risk Start?



As illustrated in the figure on the right, a risk starts when a commitment is made. For example, every time I cross the street, I run the risk of being hit by a car. The risk does not start until I commit and step into the street.

We don't have the risks of ACME delivering a low reliability subcontracted component or of ACME being late in delivering that component until we make a commitment and select ACME as our subcontractor. If we choose instead to build the component in-house, different risks exist.

We don't have the risk of a requirement not being feasible until we commit to include that requirement in our product.

When Does a Risk End?

A risk ends when one of two things happen:

One -- Bang!! I get hit by a car in the middle of the street – now it's a problem! It's no longer a risk.

Other examples where the risk turns into a problem and stops being a risk include:

- We receive delivery of a low-reliability component from ACME - problem
- ACME is late delivering the subcontracted component - problem
- We cannot figure out how to design and implement the committed to requirement - proble

Two -- I safely step onto the sidewalk on the other side of the street. The risk disappears because there is no longer the possibility of a future problem. My goal has been obtained.

Other examples where obtaining the goal removes the risk include:

- ACME delivers the subcontracted component on time with the required reliability - goal obtained
- We figure out an effective way to design and implement the committed to requirement ® goal obtained

Lesson Learned

Before it becomes an actual problem, "a risk is just an abstraction." It's something that may or may not become a future problem. There is a possibility that ignoring a risk won't come back to bite us. I can ignore the risk and run into the street over and over again without looking both ways first and never have a problem -- but then it only takes once.

What's the old saying – "it's better to be smart than lucky." Risk management is all about being smart.