

Risk Management for Software: Learning to Contain, Mitigate and Manage the Uncertainties of Software Development

Duration

2 days

Instructor

Tim Lister

Class Limit

20 students

Prerequisite

None

Price

On-site:

Please contact SPC for pricing (contact information on page 2)

Public Training:

\$995 (2 days)

*Discount available for early registration

Materials Provided

- Student manual containing the course slides
- Student handouts with class exercises and class studies

Building and maintaining software is a risky business. Since software permeates and controls so much of the present-day enterprise (and its products), lateness, cost excess and failure to perform can have far-reaching consequences. A common response to such risk is to ignore it entirely. We justify this as “positive attitude,” the heart and soul of a Can-Do management philosophy. But when real risks turn into real problems and send our projects down in flames, we can see that our past “positive attitude” was little more than Denial. There must be a better way.

The purpose of this seminar is to prepare participants to apply the budding discipline of Risk Management to software efforts. They will learn to identify and quantify the specific uncertainties that threaten success. For each uncertainty so identified, participants will learn to contain, mitigate or eliminate its impact

Welcome to the Real World!

Good luck on your next software project . . . but don't count on it. A healthy, positive attitude should allow you to expect a few decent breaks along the way, but expecting no bad breaks at all just doesn't make sense. It doesn't make sense, yet it is more or less the standard today in managing software efforts. We plan our projects based on tasks that must be done, without so much as a nod of the head toward tasks that might have to be done. Such plans reflect how the effort would proceed if nothing ever went wrong. No wonder we're plagued with time and cost overruns and quality shortfalls. Something always goes wrong.

Planning the core activities, the must-be-dones of software development, is a necessary but not sufficient beginning. Since projects never run exactly to optimal plan, we also need Risk Management. Risk Management is project management for adults. It focuses your attention constructively on the very aspects that, ignored, could lead to project debacle.

Running Away from Risk is a No-Win Strategy

A strategy of risk aversion leads us to become more and more efficient and doing things that are less and less worth doing. The projects that deliver real benefit are bound to be full of risk. Instead of running away from risk, we need to school ourselves to run toward it . . . but very, very carefully.

Since high benefit endeavors are always risky, we have to develop ways to discover the lurking risks, estimate their impact, optimize our response, and monitor for change. These are the essential skills of Risk Management.

The Opposite Of Risk Management Is Reckless Management

Just exactly what is it that we manage on software projects if not the risks? Project management IS risk management. The risk aware manager can show you a substantial list of causal risks (not just the end-result risks of lateness or overrun, but the root causes of such outcomes). He/she can tell you the likely cost in time or money should the risk materialize, and point to a specific set of materialization indicators and contingency plans to protect the project. In the absence of such sensible Risk Management, those who are counting on project success have no better strategy than to cross their fingers and hope for the best.

TRAINING



Risk Management for Software

Intended Audience

This course is intended for software managers and senior developers, and anyone sharing responsibility for success in the presence of risk

Instructor

Tim Lister has 30 years of professional software development experience and is co-author, with Tom DeMarco, of the immensely popular course sequences, Software Risk Management, Achieving Best in Class: Leading, Managing and Coaching the Modern Software Organization, and Controlling Software Projects: Management, Measurement, and Estimation. Tim Lister and Tom DeMarco are also co-authors of Peopleware: Productive Projects and Teams, 2nd ed. (Dorset House, 1999). This book has been a book club selection of the Library of Computer and Information Sciences, and has been translated into seven languages. Tim Lister and Tom DeMarco are also co-editors of Software State-of -the-Art: Selected Papers, a collection of 31 of the best papers on software published in the 1980's (Dorset House, 1990). The two partners have also produced a video entitled Productive Teams, also available through Dorset House.

Tim Lister lives in Manhattan. He holds an A.B. from Brown University, and is a member of the I.E.E.E. and the A.C.M.

For more information on this or other SPC Springboard courses, please visit www.spcspringboard.com or e-mail SPC at info@spc.ca

Software Productivity Center
Suite 460—1122 Mainland Street
Vancouver, BC V8M 4T8
www.spc.ca

Toll Free:	Fax:
1.877.548.1948	604.689.0141
Vancouver:	Toronto:
604.662.8181	416.885.0512

Outline

The statistical inevitability of risk. The link between risk and opportunity. Creating a no-fault attitude toward risk. Managing software projects by managing their risks. Risk aversion. Building a risk base. Role of the postmortem. The risks of Denver International Airport as an example.

The How-to's of Risk Management

Building a census of risks. Separating resultant and root causal risks. Object analysis of risk. Quantitative analysis of each risk. Exposure calculation. Tracking for transitions. Mitigation

strategies. Simple scheme for risk declaration.

Pre-empting Risks

Three steps to risk discovery. Roles during the discovery process. Performing backward root cause analysis. Making risk discovery safe for all hands. Alternate strategies: the spiral Win-Win model, risk and Win-Win conflicts, linking risks and requirements.

Underpinnings

Defining and calculating exposure for risks in sets. Time and cost exposure. Interpreting exposure as an indication of project health. Independence of risks. Dealing with non-binary risks Modified risk declaration scheme (with both binary and non binary risks) A modified syntax of risk declaration.

Tools and Procedures

Spreadsheet methods. VBA tools. Survey of available commercial tools. Starter toolkit for risk management.

Five Major Risks and How to Manage Them

The common risks of software development efforts, quantified and applied. The beginners' Risk Management database. Sources of tailored risk data.

Conflict and Conflict Resolution

Conflict as a source of risk in software projects. Building conflict detection and resolution skills. Common conflict scenarios and how they play out. Negotiation and mediation. Mediation within the project. Leading from a position of no power.

Bringing Risk Management to Your Organization

Coping with the risks of risk identification. Using risk analysis to tailor project plans and to form a modified process. Safety issues. Risk Management and process improvement.

A Dynamic of Risk Management

Using a modified Earned Value metric based on successive builds to track closure. Risk blocking. Progressive commitment of risk mitigation cost. Mid- and late-project de-rail strategies.



TRAINING