

CSC 7003 : Basics of Software Engineering

J Paul Gibson, A207

`paul.gibson@int-edu.eu`

<http://www-public.it-sudparis.eu/~gibson/Teaching/CSC7003/>

Risk Management in Software Engineering

<http://www-public.it-sudparis.eu/~gibson/Teaching/CSC7003/L11-Risks.pdf>

Some Background Reading – Critical Analysis

- Barry W. Boehm, "Software Risk Management: Principles and Practices," *IEEE Software*, pp. 32-41, January/February, 1991
- Richard Fairley. 1994. Risk Management for Software Projects. *IEEE Softw.* 11, 3 (May 1994), 57-67. DOI=10.1109/52.281716 <http://dx.doi.org/10.1109/52.281716>
- Higuera, Ronald P. and Haimes, Yacov Y. *Software Risk Management* (CMU/SEI-96-TR-012). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, June 1996.
- Mark Keil, Paul E. Cule, Kalle Lyytinen, and Roy C. Schmidt. 1998. A framework for identifying software project risks. *Commun. ACM* 41, 11 (November 1998), 76-83. DOI=10.1145/287831.287843 <http://doi.acm.org/10.1145/287831.287843>
- Ropponen, J.; Lyytinen, K.; , "Components of software development risk: how to address them? A project manager survey," *Software Engineering, IEEE Transactions on* , vol.26, no.2, pp.98-112, Feb 2000, doi: 10.1109/32.841112

Some Background Reading – Critical Analysis

These are some of the most cited publications on risk management in software development.

Local copies are downloadable from the module web site.

Each project team is to spend 30 minutes browsing the publications looking for advice that they think would help them with managing risk in their 15-puzzle development

Each team is to summarize their findings in a 15 minute presentation – your findings may be negative as well as positive.

NOTE: You will study this in more detail in the management module that follows later this year.

Software

Software Risk Management: Principles and Practices

January/February 1991 (vol. 8 no. 1)

pp. 32-41

Barry W. Boehm

DOI Bookmark: <http://doi.ieeecomputersociety.org/10.1109/52.62930>

ABSTRACT

The emerging discipline of software risk management is described. It is defined as an attempt to formalize the risk-oriented correlates of success into a readily applicable set of principles and practices. Its objectives are to identify, address, and eliminate risk items before they become either threats to successful software operation or major sources of software rework. The basic concepts are set forth, and the major steps and techniques involved in software risk management are explained. Suggestions for implementing risk management are provided.

ADDITIONAL INFORMATION

Index Terms:

software engineering; software risk management; software rework; DP management; software engineering

Citation:

Barry W. Boehm, "Software Risk Management: Principles and Practices," *IEEE Software*, vol. 8, no. 1, pp. 32-41, Jan./Feb. 1991, doi:10.1109/52.62930

Risk management for software projects



Fairley, R.;

Software Eng. Manage. Assoc., Woodland Park, CO, USA

This paper appears in: [Software, IEEE](#)

Issue Date: May 1994

Volume: 11 **Issue:**3

On page(s): 57 - 67

ISSN: 0740-7459

References Cited: 4

INSPEC Accession Number: 4708188

Digital Object Identifier: [10.1109/52.281716](#)

Date of Current Version: 06 August 2002

Sponsored by: [IEEE Computer Society](#)

ABSTRACT

There is little to instruct software project managers on how to handle risk in a way that ensures the success of contingency planning and avoids crisis. This seven-step procedure describes how to identify risk factors, calculate their probability and effect on a project, and plan for and conduct risk management



Library

Seminal works and reference material created by SEI staff.

Front Desk Browse the Stacks

Software Risk Management


There are many problems associated with requirements engineering, including problems in defining the system scope, problems in fostering understanding among the different communities affected by the development of a given system, and problems in dealing with the volatile nature of requirements. These problems may lead to poor requirements and the cancellation of system development, or else the development of a system that is later judged unsatisfactory or unacceptable, has high maintenance costs, or undergoes frequent changes. By improving requirements elicitation, the requirements engineering process can be improved, resulting in enhanced system requirements and potentially a much better system.



PDF [209 KB]

Authors

A framework for identifying software project risks

Full Text:  [Pdf](#)

Authors: [Mark Keil](#) Georgia State Univ., Atlanta
[Paul E. Cule](#) Marquette Univ., Milwaukee, WI
[Kalle Lyytinen](#) Univ. of Jyväskylä, Finland
[Roy C. Schmidt](#) Bradley Univ., Peoria, IL



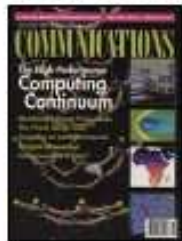
1998 Article



[Bibliometrics](#)

- Downloads (6 Weeks): 110
- Downloads (12 Months): 969
- Citation Count: 58

Published in:



· Magazine

Communications of the ACM [CACM Homepage](#) [archive](#)

Volume 41 Issue 11, Nov. 1998

[ACM](#) New York, NY, USA

[table of contents](#) doi>[10.1145/287831.287843](https://doi.org/10.1145/287831.287843)

Components of software development risk: how to address them? A project manager survey



Ropponen, J.; Lyytinen, K.;
Finnish Evangelical Lutheran Mission, Helsinki

This paper appears in: [Software Engineering, IEEE Transactions on](#)

Issue Date: Feb 2000

Volume: 26 **Issue:** 2

On page(s): 98 - 112

ISSN: 0098-5589

References Cited: 48

INSPEC Accession Number: 6612331

Digital Object Identifier: [10.1109/32.841112](#)

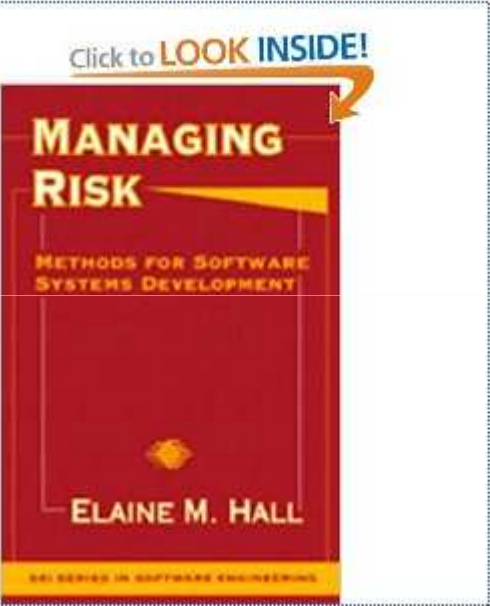
Date of Current Version: 06 August 2002

Sponsored by: [IEEE Computer Society](#)

ABSTRACT

Software risk management can be defined as an attempt to formalize risk oriented correlates of development success into a readily applicable set of principles and practices. By using a survey instrument we investigate this claim further. The investigation addresses the following questions: 1) What are the components of software development risk? 2) how does risk management mitigate risk components, and 3) what environmental factors if any influence them? Using principal component analysis we identify six software risk components: 1) scheduling and timing risks, 2) functionality risks, 3) subcontracting risks, 4) requirements management, 5) resource usage and performance risks, and 6) personnel management risks. By using one-way ANOVA with multiple comparisons we examine how risk management (or the lack of it) and environmental factors (such as development methods, manager's experience) influence each risk component. The analysis shows that awareness of the importance of risk management and systematic practices to manage risks have an effect on scheduling risks, requirements management risks, and personnel management risks. Environmental contingencies were observed to affect all risk components. This suggests that software risks can be best managed by combining specific risk management considerations with a detailed understanding of the environmental context and with sound managerial practices, such as relying on experienced and well-educated project managers and launching correctly sized projects

SOME FUNDAMENTALS taken from (inspired by) *Managing Risk: Methods for Software Systems Development* by Elaine M. Hall, Addison-Wesley 1998



Click to **LOOK INSIDE!**

MANAGING RISK
METHODS FOR SOFTWARE SYSTEMS DEVELOPMENT
ELAINE M. HALL
SERIES IN SOFTWARE ENGINEERING

Managing Risk: Methods for Software Systems Development [Hardcover]
Elaine M. Hall (Author)
★★★★☆ (8 customer reviews) (0)


Price: **\$69.99** & this item ships for **FREE with Super Saver Shipping**. [Details](#)

In Stock.
Ships from and sold by **Amazon.com**. Gift-wrap available.

Only 2 left in stock--order soon (more on the way).

Want it delivered Wednesday, January 19? Order it in the next **14 hours and 3 minutes**, and choose **One-Day Shipping** at checkout. [Details](#)

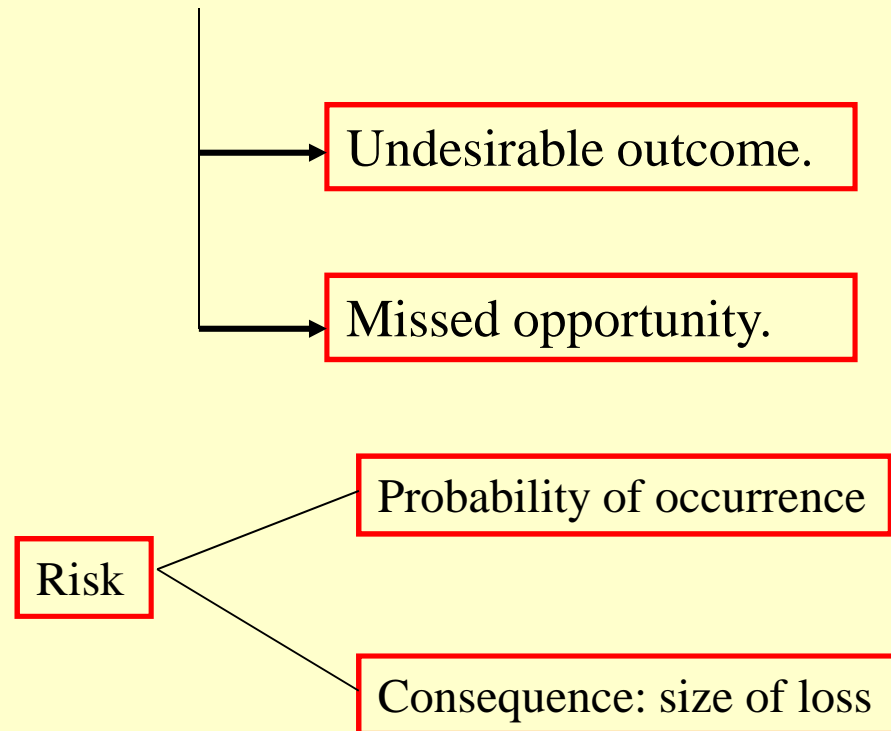
21 new from \$17.99 **34 used** from \$1.37

 **FREE Two-Day Shipping for Students.** [Learn more](#)

[Share your own customer images](#)
[Look inside this book](#)

FOR DISCUSSION: What is risk?

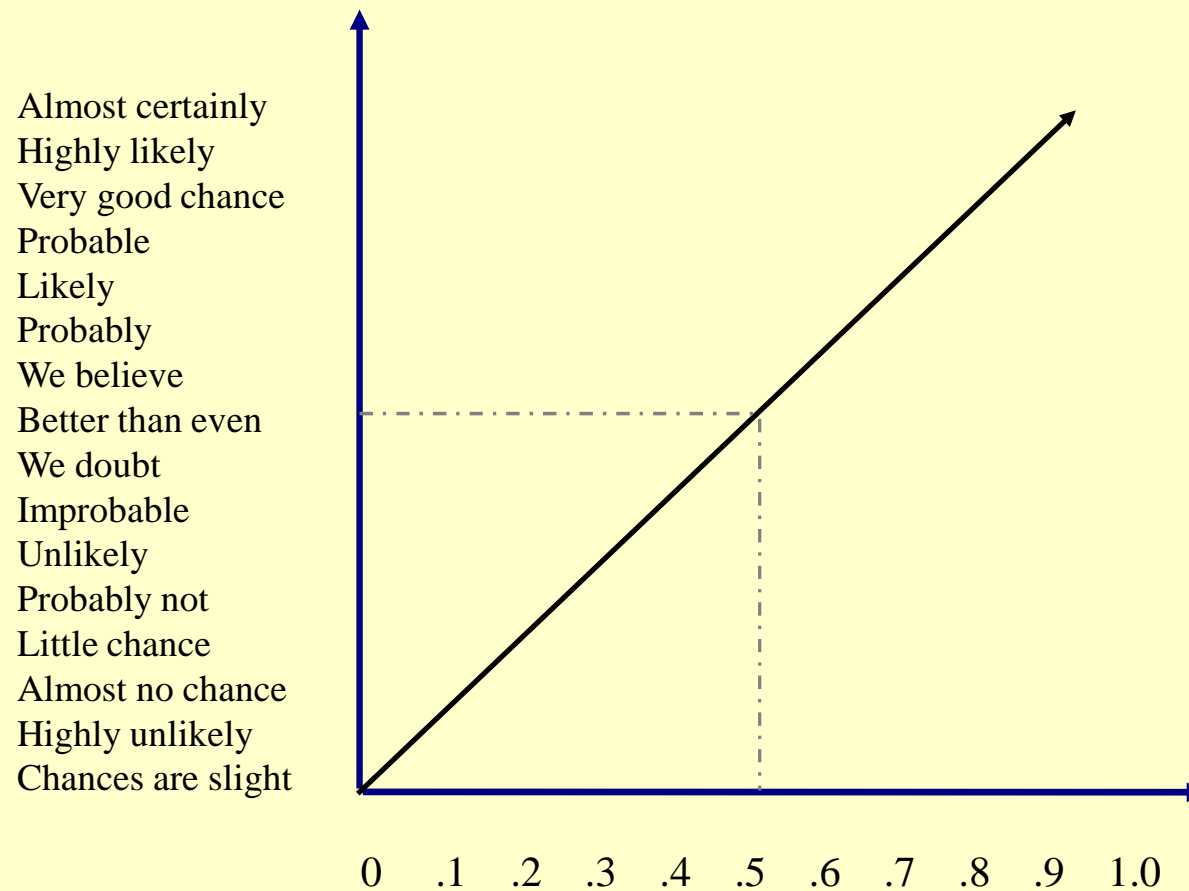
A risk is a possibility of loss.



Question: what is a reasonable unit of measurement for exposure

Risk Exposure = Probability x Consequence

Perceived Probability



Consequence : what units of measurement?

- Time
- Cost
- Morale
- ...?

Question: Can these consequences be normalized? Should they be?

Classification of software risks

There are many different classification frameworks, eg:

- Software Project Risks
 - Resource constraints, external interfaces, supplier relationships, nonperforming vendors, internal politics, interteam/intergroup coordination problems, inadequate funding.
- Software Process Risks
 - Undocumented software process, lack of effective peer reviews, no defect prevention, poor design process, poor requirements management, ineffective planning.
- Software Product Risks
 - Lack of domain expertise, complex design, poorly defined interfaces, poorly understood legacy system(s), vague or incomplete requirements.

Most classification frameworks add details to the generic classes.

For example:

SOFTWARE REQUIREMENT RISKS

Lack of analysis for change of requirements	Change extension of requirements
Lack of report for requirements	Poor definition of requirements
Ambiguity of requirements	Change of requirements
Inadequate of requirements	Impossible requirements
Invalid requirements	

SOFTWARE COST RISKS

Lack of good estimation in projects	Unrealistic schedule
The hardware does not work well	Human errors
Lack of testing	Lack of monitoring
Complexity of architecture	Large size of architecture
Extension of requirements change	The tools does not work well
Personnel change	Management change
Technology change	Environment change
Lack of reassessment of management cycle	

SOFTWARE SCHEDULING RISKS

Inadequate budget	Change of requirements
Extension of requirements change	Human errors
Inadequate knowledge about tools	Inadequate knowledge about techniques
Long-term training for personnel	Lack of employment of manager experience
Lack of enough skill	Lack of good estimation in projects
Lack of accurate system domain definition	Lack of goals specification
Difficulty of implementation	Disagreement between members
Lack of tools	Shortage of personnel
Tools failure	Technology change
Lack of agreement between customer and developer	Slow management cycle
Supply budget in inappropriate time	Environment change
Lack of a good guideline	

SOFTWARE QUALITY RISKS

Inadequate documentation	Lack of project standard
Lack of design documentation	Inadequate budget
Human errors	Unrealistic schedule
Extension of requirements change	Poor definition of requirements
Lack of enough skill	Lack of testing
Lack of good estimation in projects	Inadequate knowledge about techniques
Lack of employment of manager experience	Lack of accurate system domain definition
The simulator is to be destroyed	Lack of reassessment
Inadequate knowledge about programming language	Inadequate knowledge about tools
The hardware does not work well	Lack of analysis for change of requirements
The tools do not work well	Loss technical equipment
Lack of stability between personnel	Personnel change
Weakness of management	Lack of commitment
Disagreement between members	Ambiguity of requirements
Complexity of architecture	Incomplete requirements
Lack of roles and responsibilities definition	Inadequate training of personnel
Management change	Technology change
Lack of collaboration between developer	Environment change
Lack of a good guideline	

SOFTWARE BUSINESS RISKS

The products that no one want them

The products that are not suitable with total strategy

The products that sellers do not know how to sell them

Failure in total budget

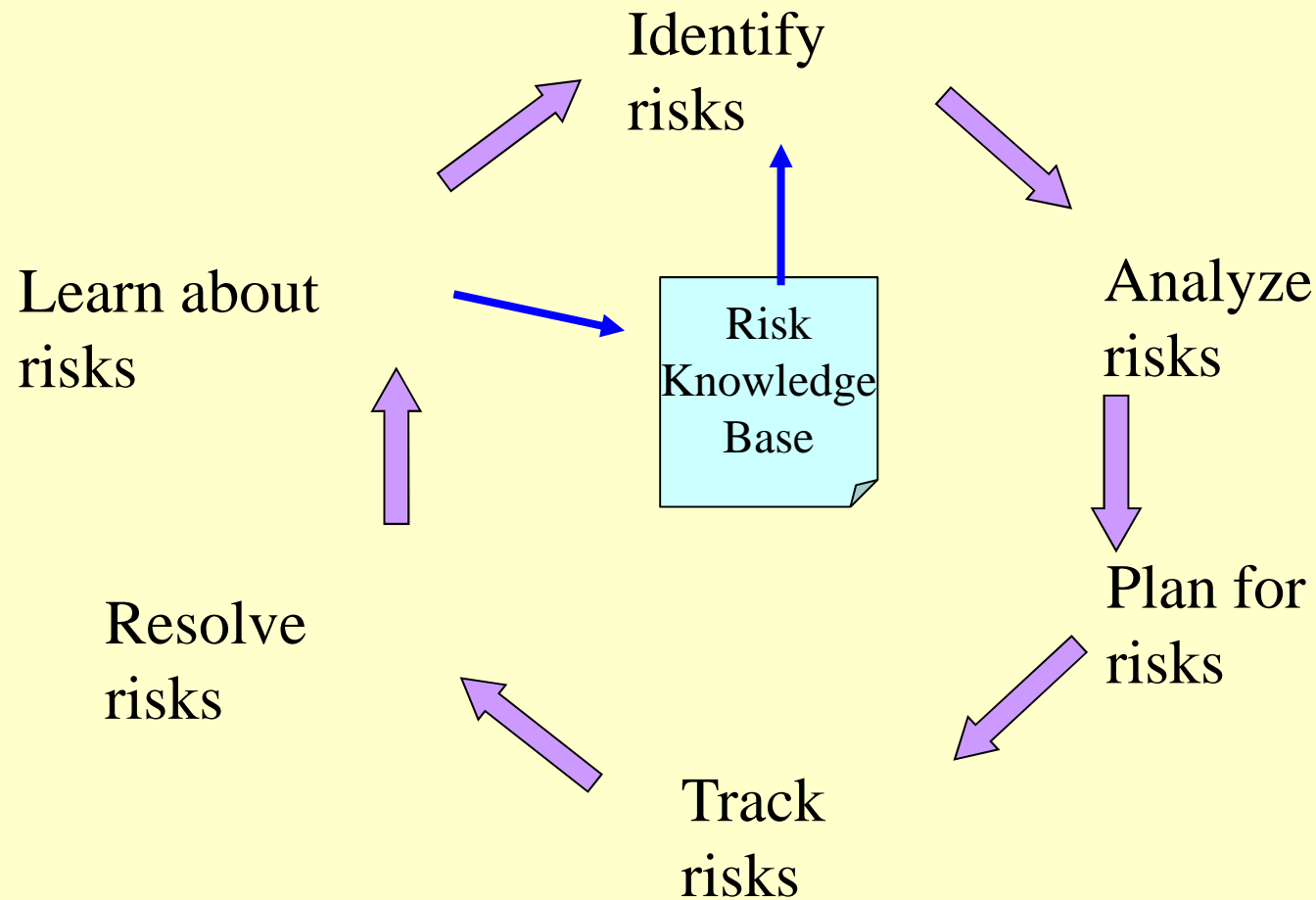
Failure in commitment

Failure in management because of change in different people

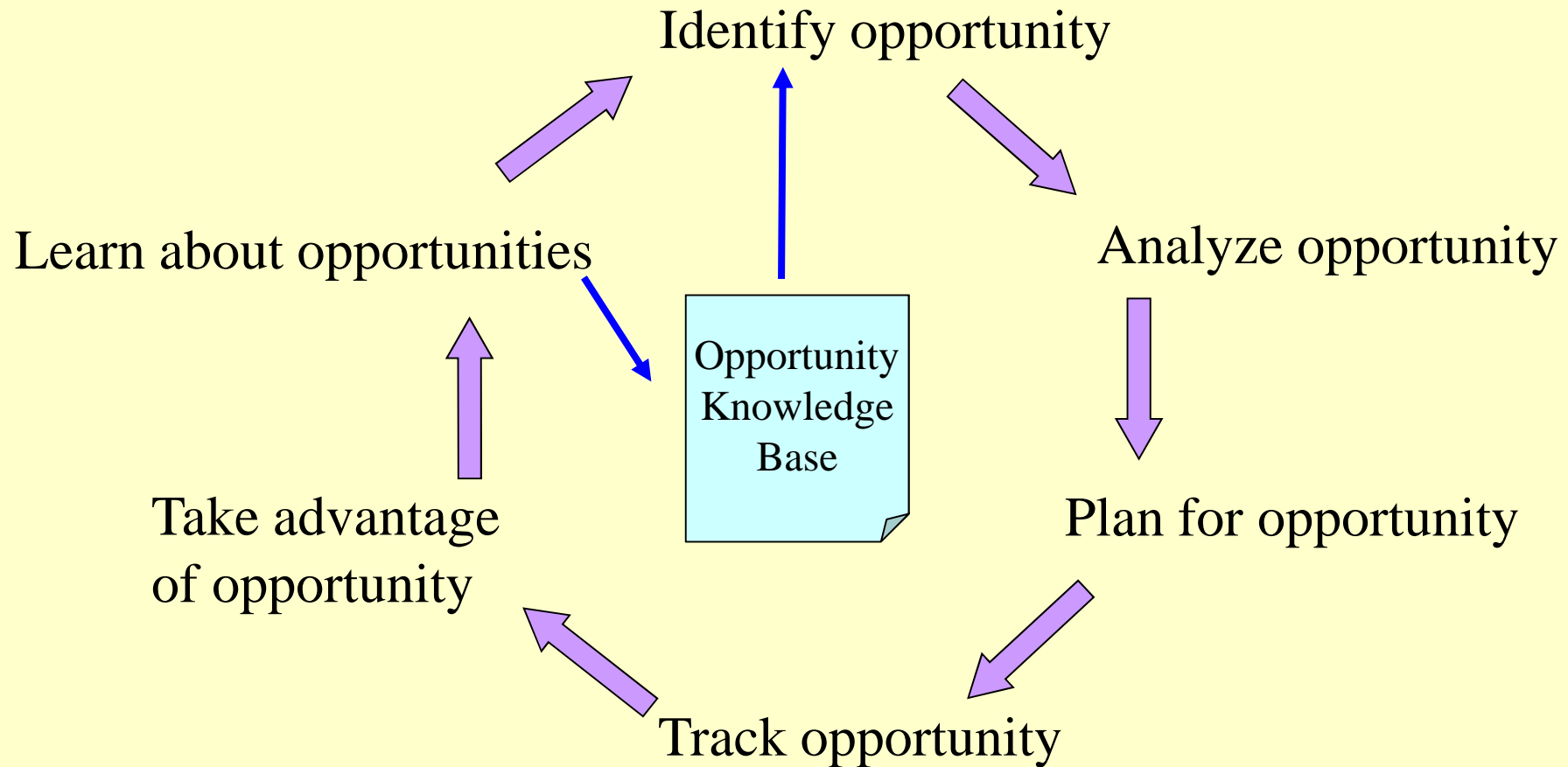
NOTE: some structures/frameworks ignore/overlook/abstract away from certain (classes of) risks.

For example: what if there is a risk that you will lose key staff/ideas

The Risk Management Process



Opportunity Management: same structure as risk management



Note: we will focus on risks

Identification: Documentation

Header
Assessment
Action Plan
Tracking
Resolution

Project	<i>Name of project</i>
Date	<i>Date of entry</i>
Risk name	<i>Name of risk</i>
Risk category	<i>Type of risk</i>
Probability	<i>Likelihood of occurrence</i>
Consequence	<i>Severity of impact</i>
Originator	<i>Who reported this risk</i>
Phase/activity	<i>Where in software process</i>
WBS Element	<i>WBS relationship</i>

Header
Assessment
Action Plan
Tracking
Resolution

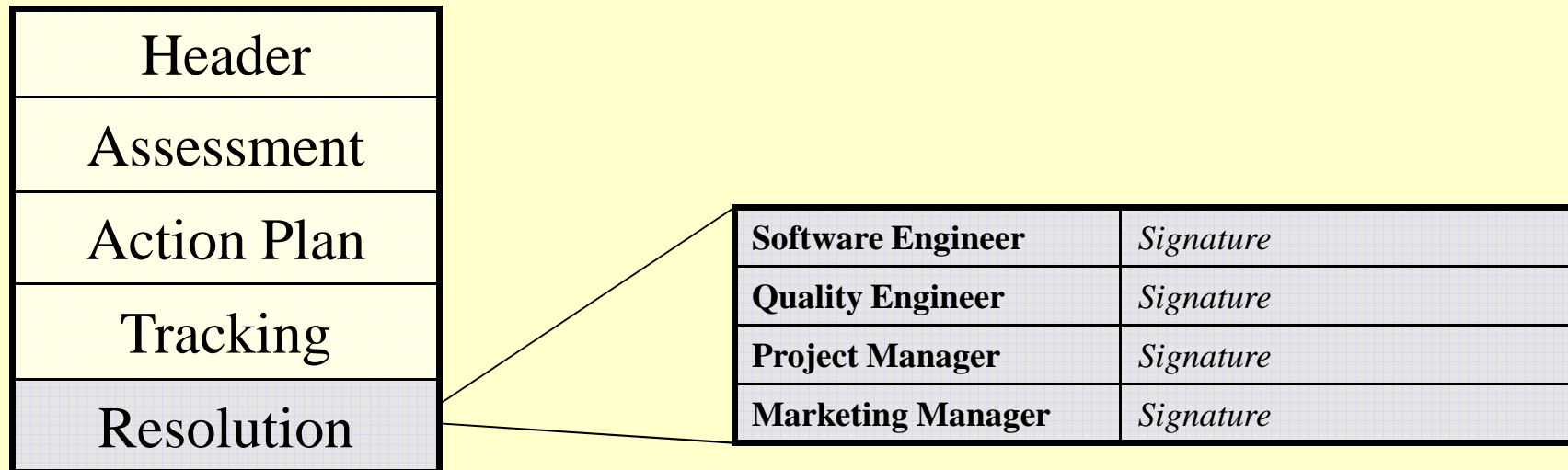
Statement	<i>Brief description of risk</i>
Context	<i>When, where, how, why</i>
Analysis	<i>Impact on project</i>

Planning/Tracking: Documentation

Header
Assessment
Action Plan
Tracking
Resolution

Scenario	<i>What would happen?</i>
Indicator	<i>Metric to be monitored</i>
Trigger condition	<i>Value indicating risk scenario</i>
Checkpoint	<i>When/where to check metric</i>
Resolution strategy	<i>How we will handle the risk</i>
Action plan	<i>Concrete action plan</i>

Resolution: Documentation



Risk Management Capability

5: Risk statistics used to make organizational/process improvements

4: Quantified analysis used to determine resolution cost/benefit for project

3: Risks systematically quantified, analyzed, planned, tracked and resolved

2: Risks are usually recorded, tracked and handled as they are discovered

1: Risks ignored or only tracked in an ad-hoc fashion

NOTE: In your current project you should be aiming for level 2.