Fundamentals
of
Software Risk Management

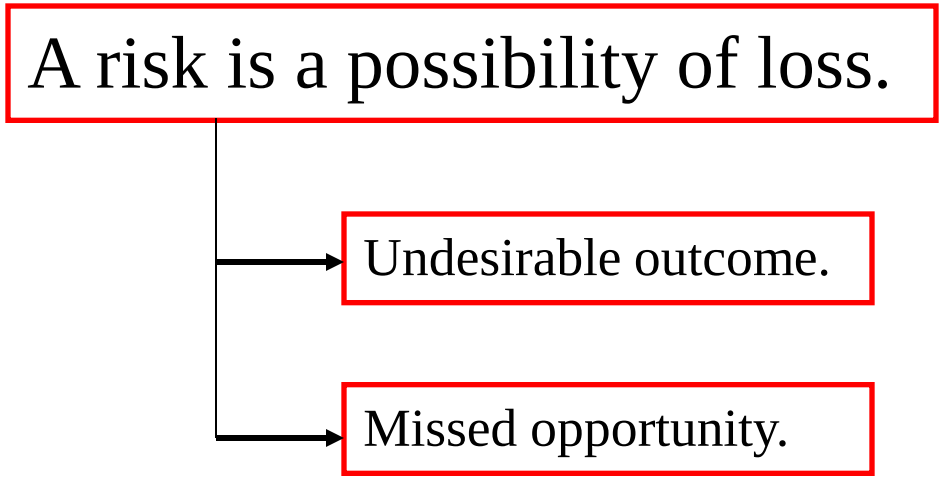
Why do software projects go wrong?

- Inadequate understanding of customer needs
- Poor requirements
- Poor requirements management
- Poor or no architecture/design
- Code first and ask questions later
- Poorly understood legacy design/code
- No peer reviews to catch problems early
- Inexperienced or incapable personnel
- Ineffective testing – misses serious defects
- …

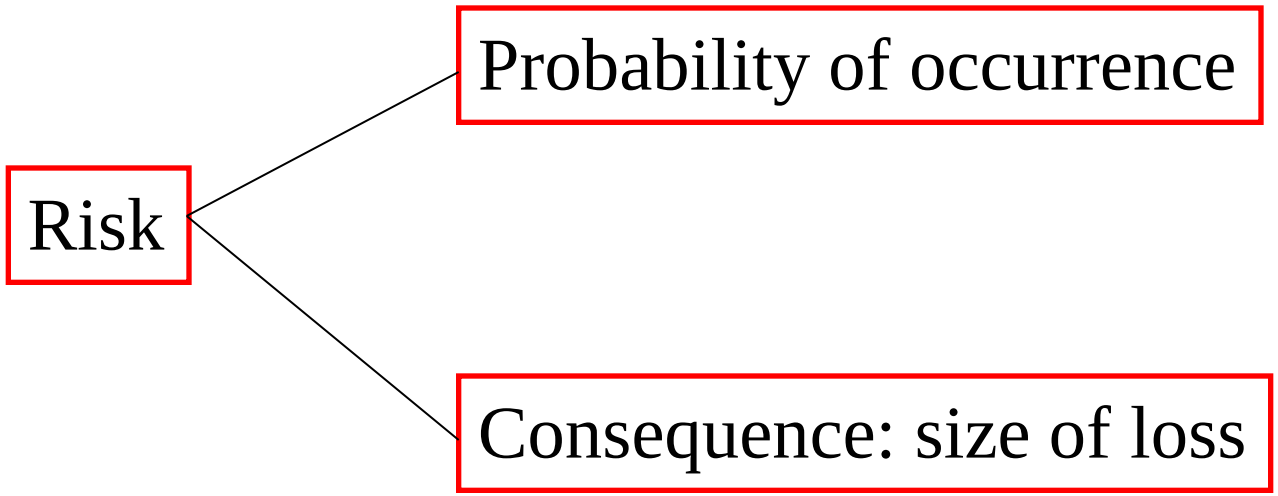
Software Risk Management

Risk Management is a practice with processes, methods, and tools for managing risks in a project.

What is risk?


A risk is a possibility of loss.
Undesirable outcome.
Missed opportunity.

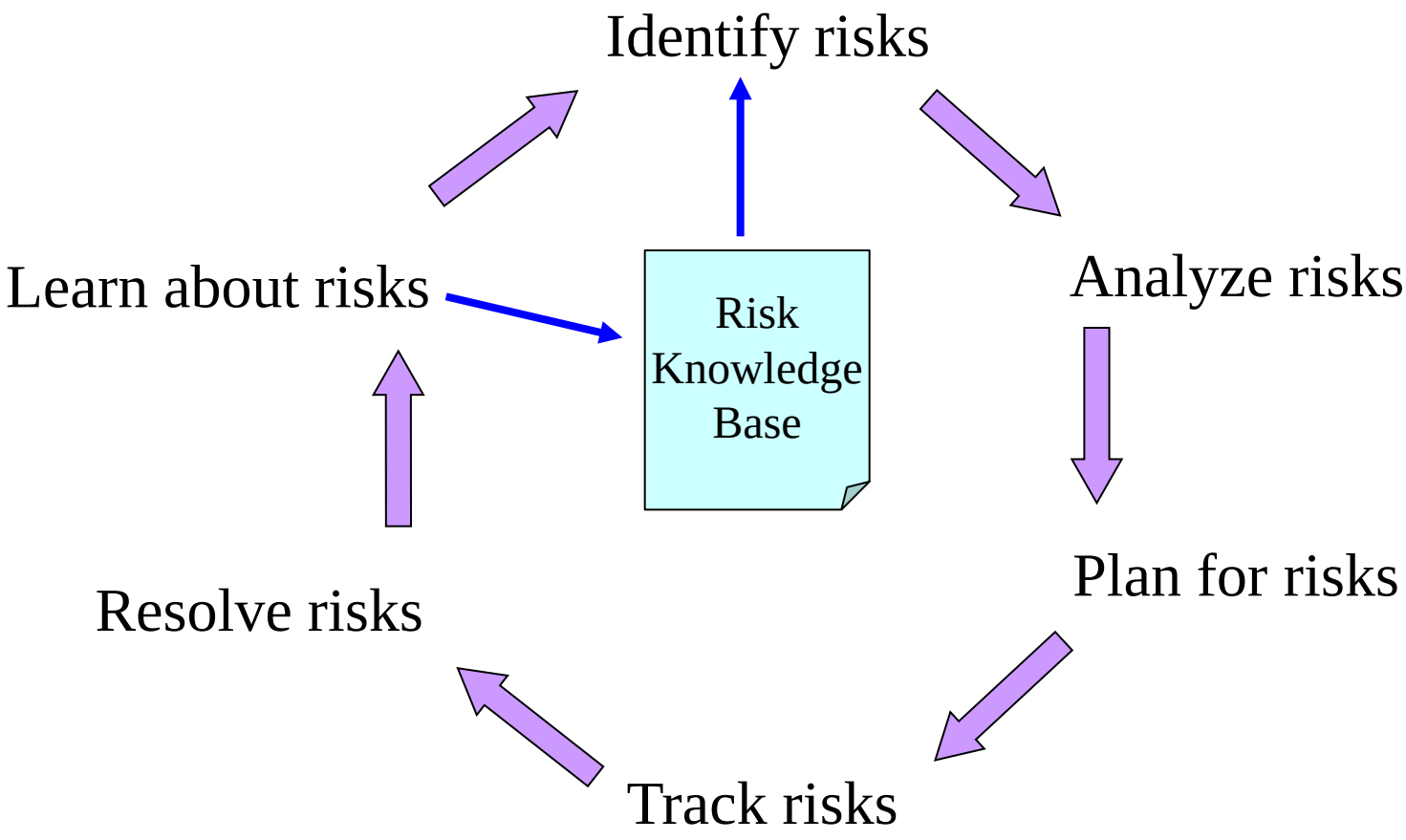
Anatomy of a risk



Classification of software risks

- 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.

The Risk Management Process

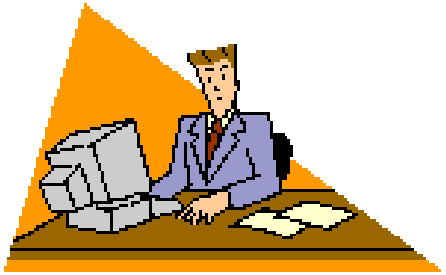


Identification: Discovery

Team Brainstorming

Call for Risks (CFR)

Walkthroughs



Spurious

Risk Knowledge Base

Identification: Quantification

Risk Exposure = Probability x Consequence

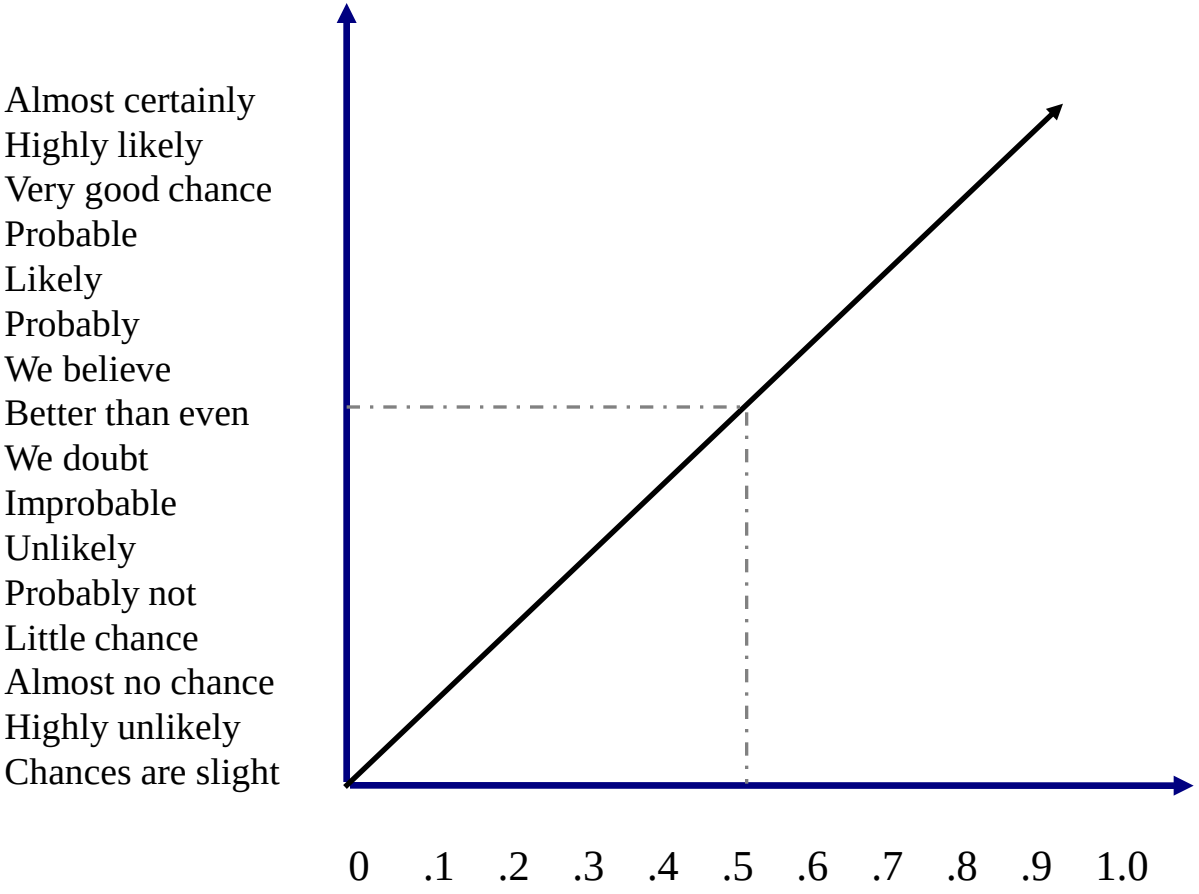
Calculating Risk Exposure

Factor	P	C	RE
Late delivery from COTS vendor ACME	0.25	28 days	7 days
ACME API integration delay	0.6	15 days	9 days
Additional unit testing needed; 3% more classes than first estimated	0.9	20 days	18 days
Beta test group reports that they may not be able to fit us into their pipeline until May 1 instead of April 1	0.5	30 days	15 days
TOTAL RISK EXPOSURE			49 days

Note: For simplicity, all risk consequences are stated in calendar days.

Perceived Probability

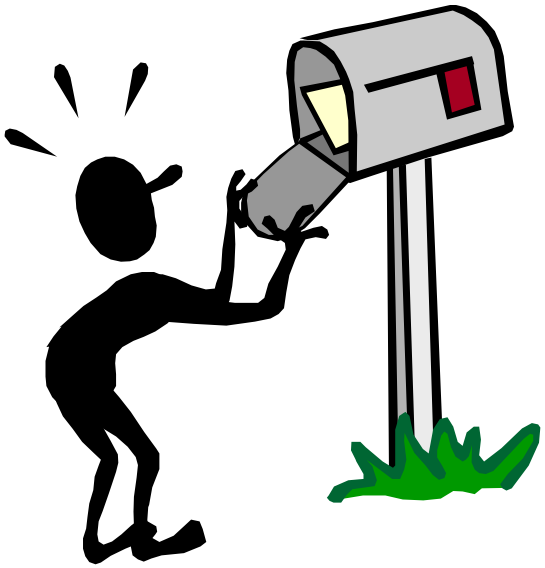
Adapted from *Managing Risk: Methods for Software Systems Development* by Elaine M. Hall, Addison-Wesley 1998



Why quantify risk

- Allows solution ideas to be evaluated more critically
- Encourages design awareness of risk
- Allows feedback on risks we missed
- Allows feedback on impact of risks we anticipated
- Allows us to allocate resources to deal with risks
- Allows us to determine whether a risk is acceptable

Identification: Communication



Notify all affected stakeholders:

- Customers
- Project/Program Manager
- Fellow Team Members
- Management
- Marketing
- Sales
- Customer Support
- Finance
- Quality Assurance
- SEPG
- …

Analysis of risks: Questions

- How severe is the consequence?
- How likely is the occurrence?
- Is the risk exposure acceptable?
- How soon must the risk be dealt with?
- What is causing the risk?
- Are there similarities between risks?
- Are there dependency relationships?
- What are the risk drivers?

Analysis of risks: Activities

- **Grouping**
  - Eliminate redundant risks; Combine related risks; Link dependent risks

- **Determining risk drivers**
  - Underlying factors that affect severity of consequence
  - May affect estimation of probability, consequence, risk exposure
  - Increases understanding of how risks can be mitigated

- **Ranking**
  - Order of likelihood, consequence, exposure, time frame

- **Determining root causes (sources of risk)**
  - Old-fashion root cause analysis,
  - Identify common root causes

Planning: Resolution Strategies

- **Risk Avoidance**
  - Prevent the risk from occurring, reduce probability to zero
- **Risk Protection**
  - Reduce the probability and/or consequence of the risk before it happens
- **Risk Reduction**
  - Reduce the probability and/or consequence of the risk after it happens
- **Risk Research**
  - Obtain more information to eliminate or reduce uncertainty
- **Risk Reserves**
  - Use previously allocated schedule or budget slack
- **Risk Transfer**
  - Rearrange things to shift risk elsewhere (to another group, for example)

Planning: Activities

- Specify scenarios
  - How would we be able to tell it is really happening?
- Define quantified threshold for early warning
  - What to monitor, when we consider the risk to be happening
- Develop resolution alternatives
  - Ways to eliminate, mitigate or handle the risk
- Select resolution approach
  - What has the best ROI?
- Specify risk action plan
  - Document decisions

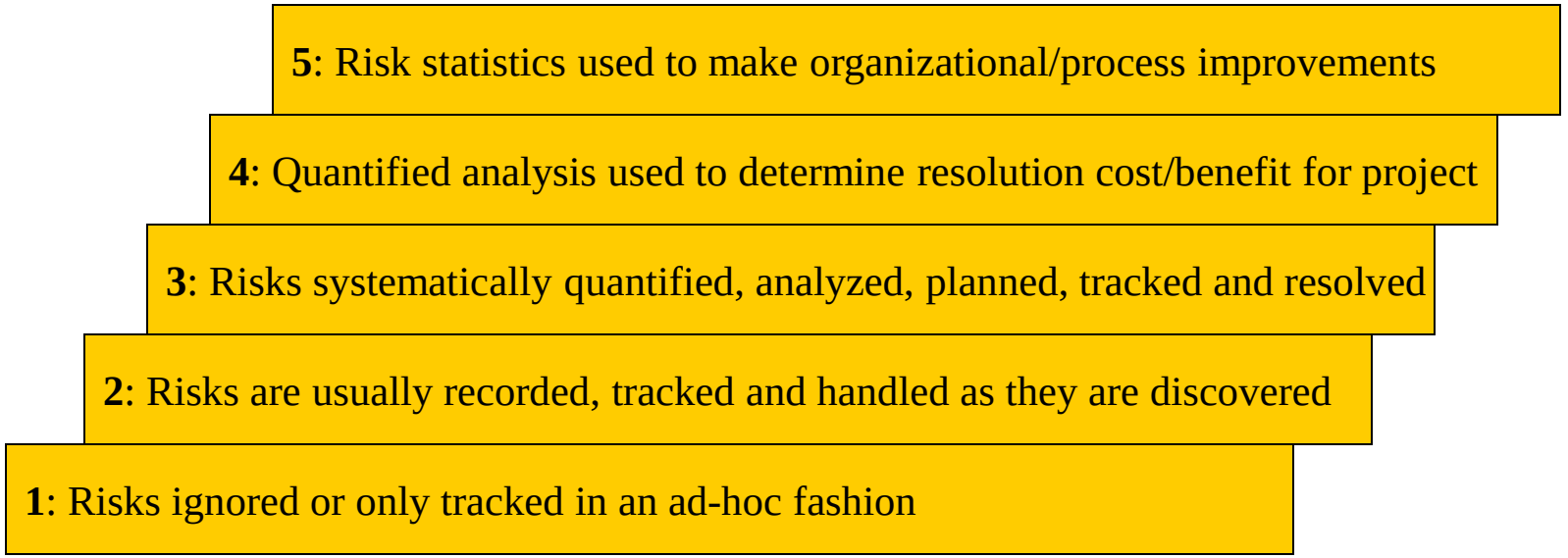
Tracking

- **Monitor risk scenarios**
  - Watch for signs of a risk scenario occurring
- **Compare indicators to trigger conditions**
  - Watch indicator metrics – do they satisfy trigger conditions?
- **Notify stakeholders**
  - Let stakeholders know the risk is happening; execute action plan
- **Collect statistics**
  - Update risk database

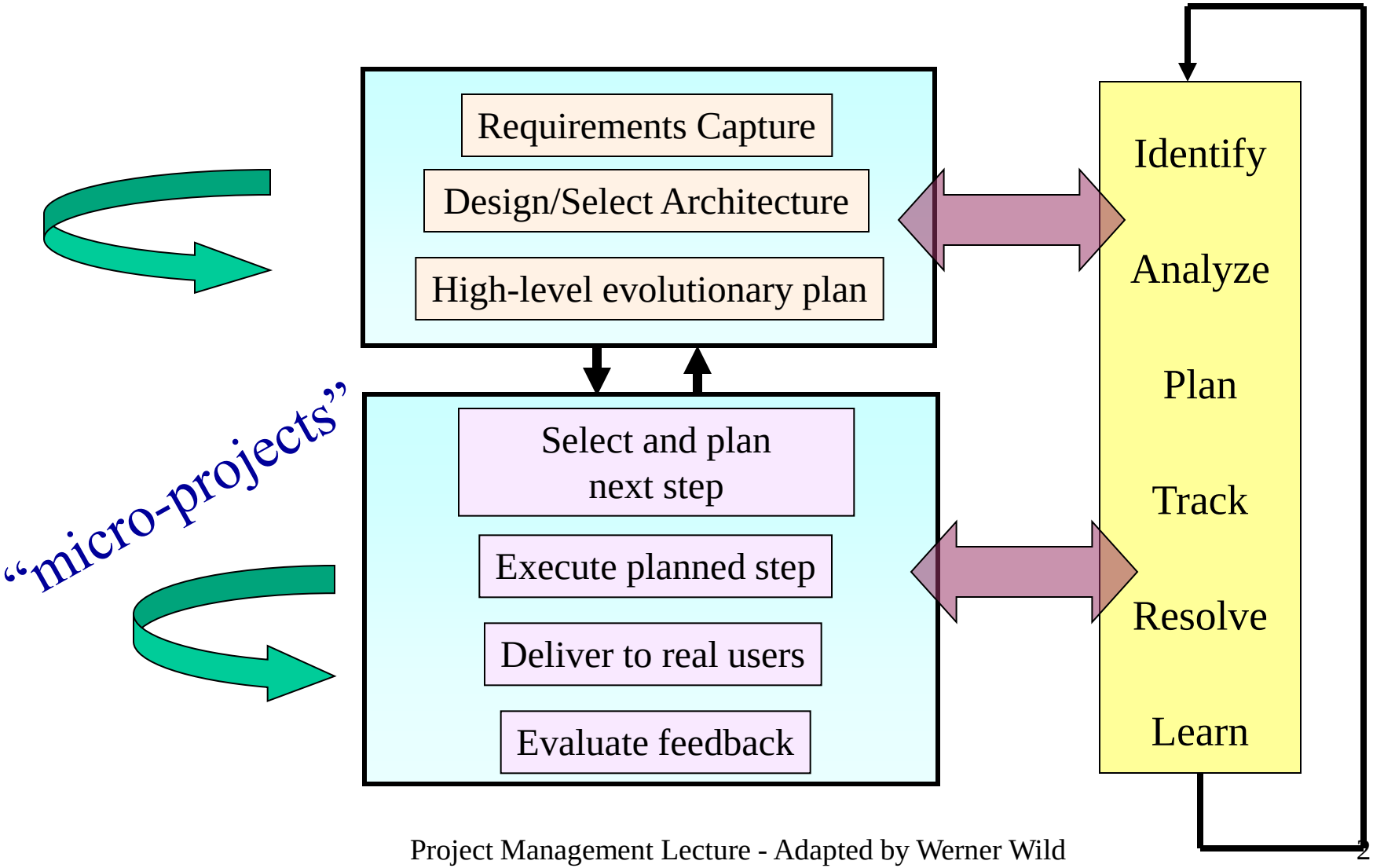
Resolution

- **Acknowledge receipt of notification**
  - Let stakeholders know you are “on the ball”
  - Indicate response time
  - Determine accountability/ownership
- **Execute action plan**
  - Improvise, adapt, overcome
  - Wanted: common sense
- **Provide continuous updates**
  - Let stakeholders know your progress in resolving the risk
- **Collect statistics**
  - Update risk database

Risk Management Capability



Evolutionary Delivery

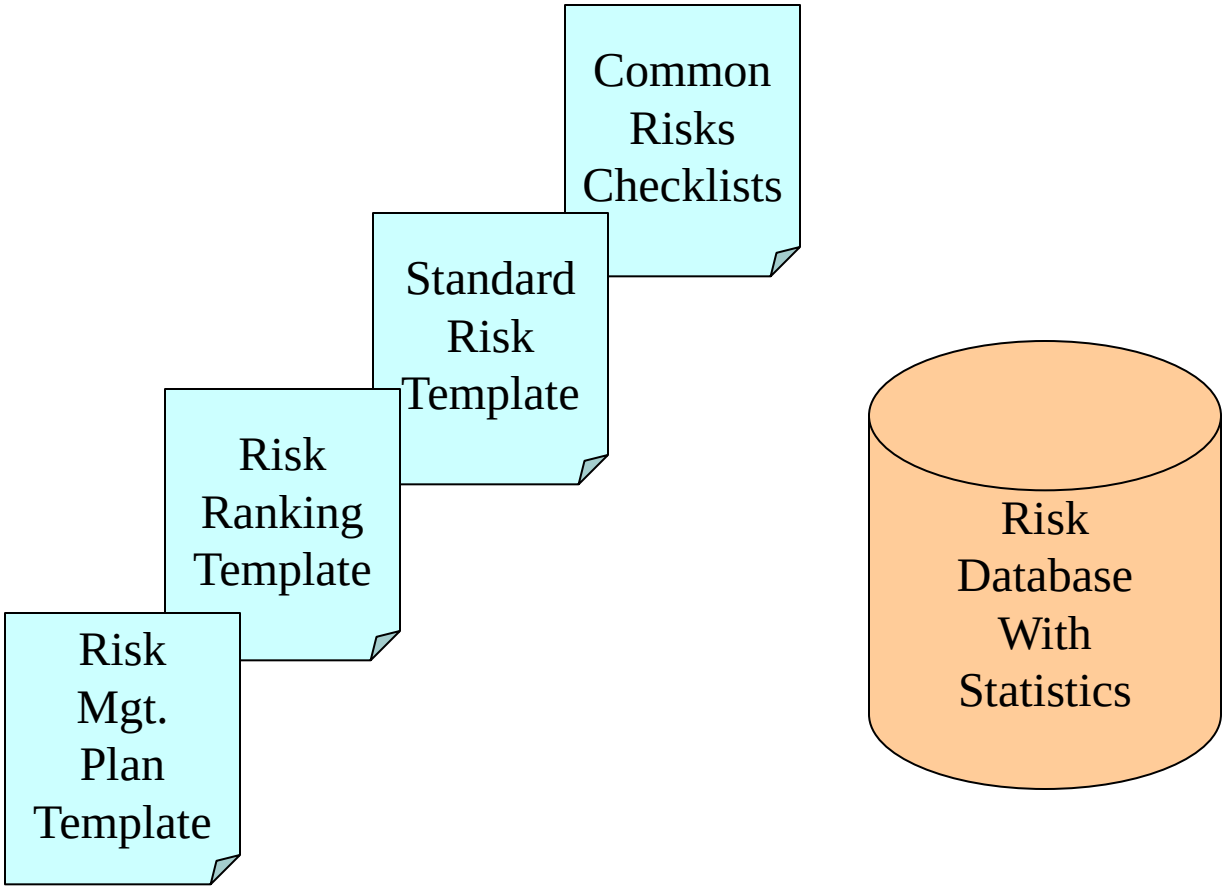


Learning from risks

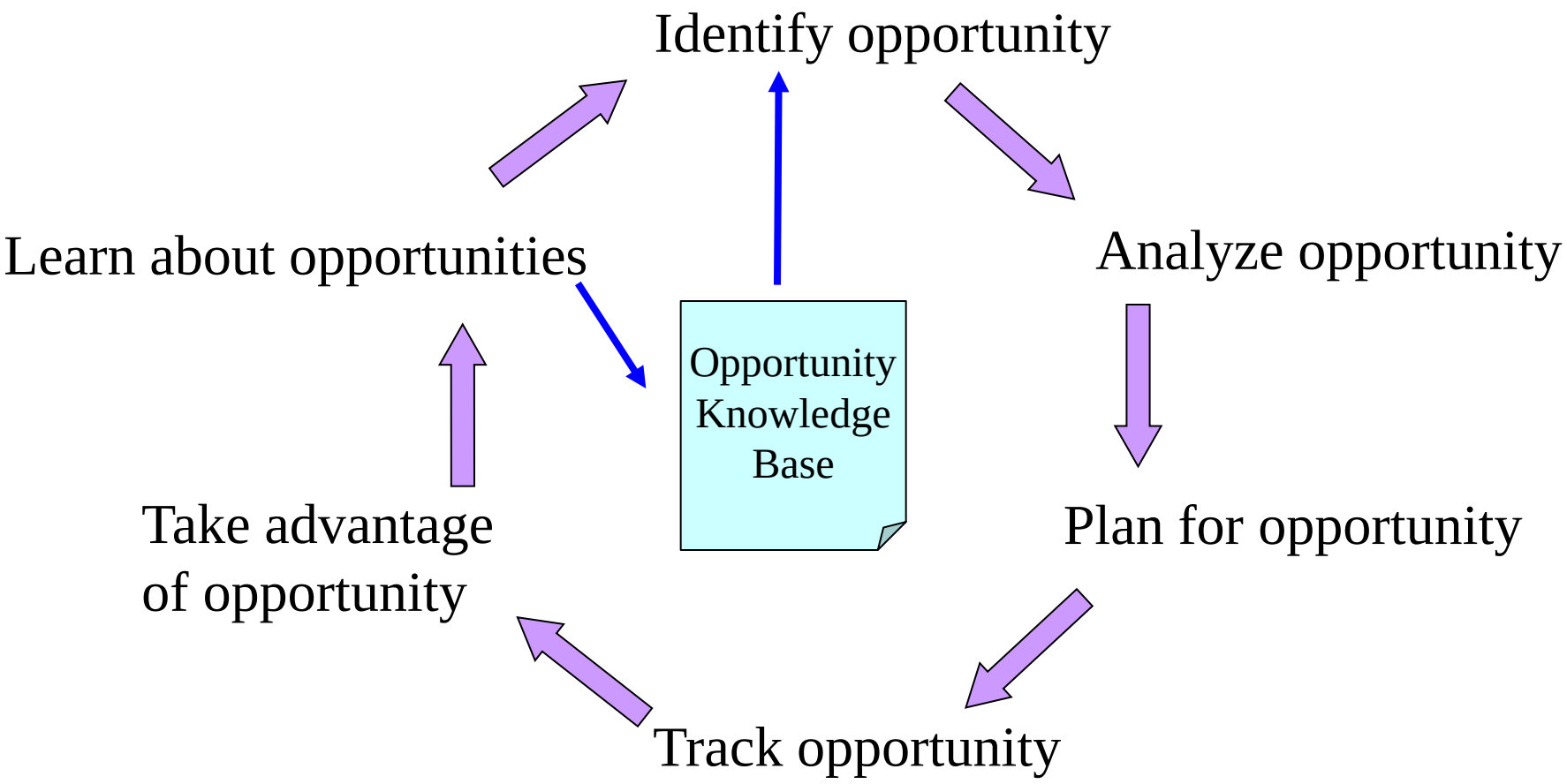
- **Project Retrospective / Post mortem:**
  - What were the unanticipated risks?
  - What was the actual severity of consequence?
  - What resolution strategies worked well/not so well?
  - What types of risks could we
    - prevent or transfer?
    - protect ourselves from or reduce?
    - handle only by allocating reserves?

- **Action:**
  - What are the preventative measures we can take in the future?
  - What can the SEPG do?
  - Are there significant vendor/partner performance problems?
  - What can we share with other project teams?

Risk Management Infrastructure



Opportunity Management



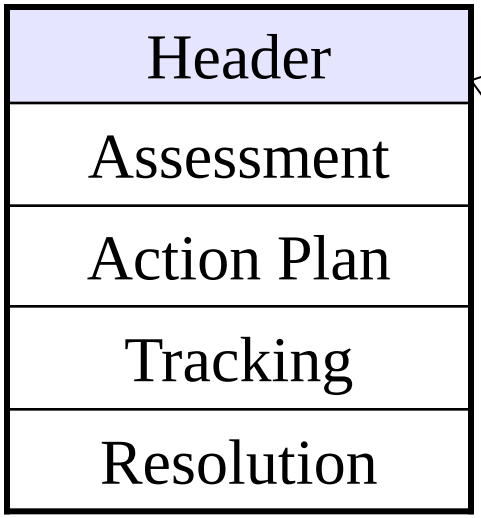
Acknowledgements

This presentation is based on one from

Ødegård Labs, Inc.

Identification: Documentation

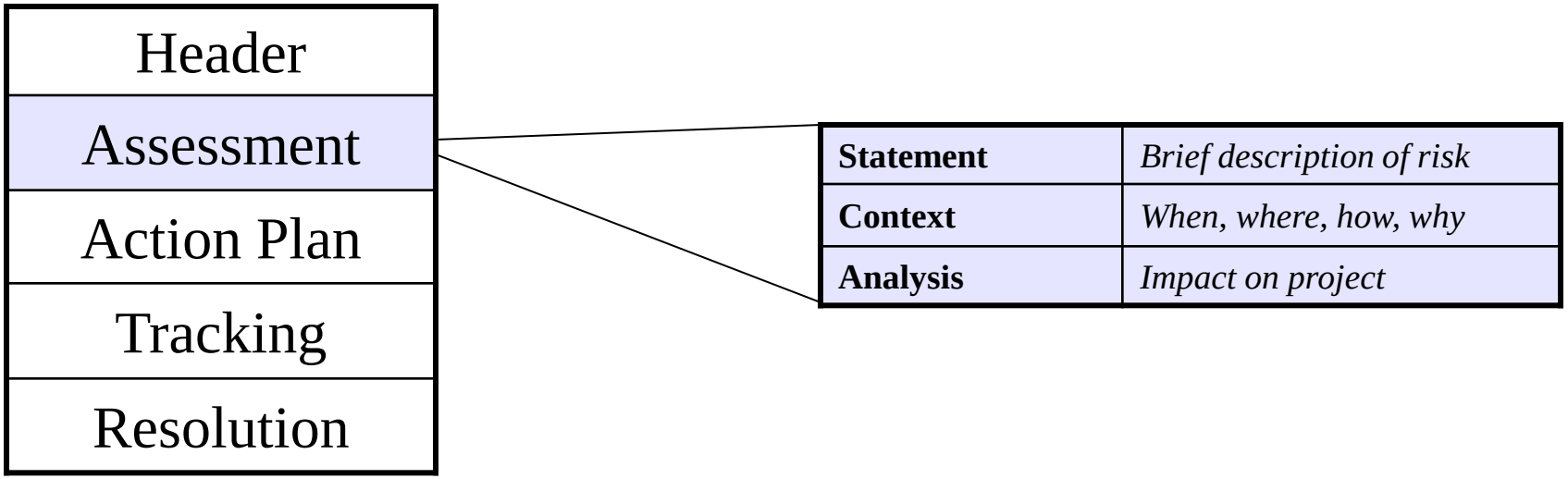
Adapted from *Managing Risk: Methods for Software Systems Development* by Elaine M. Hall, Addison-Wesley 1998



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

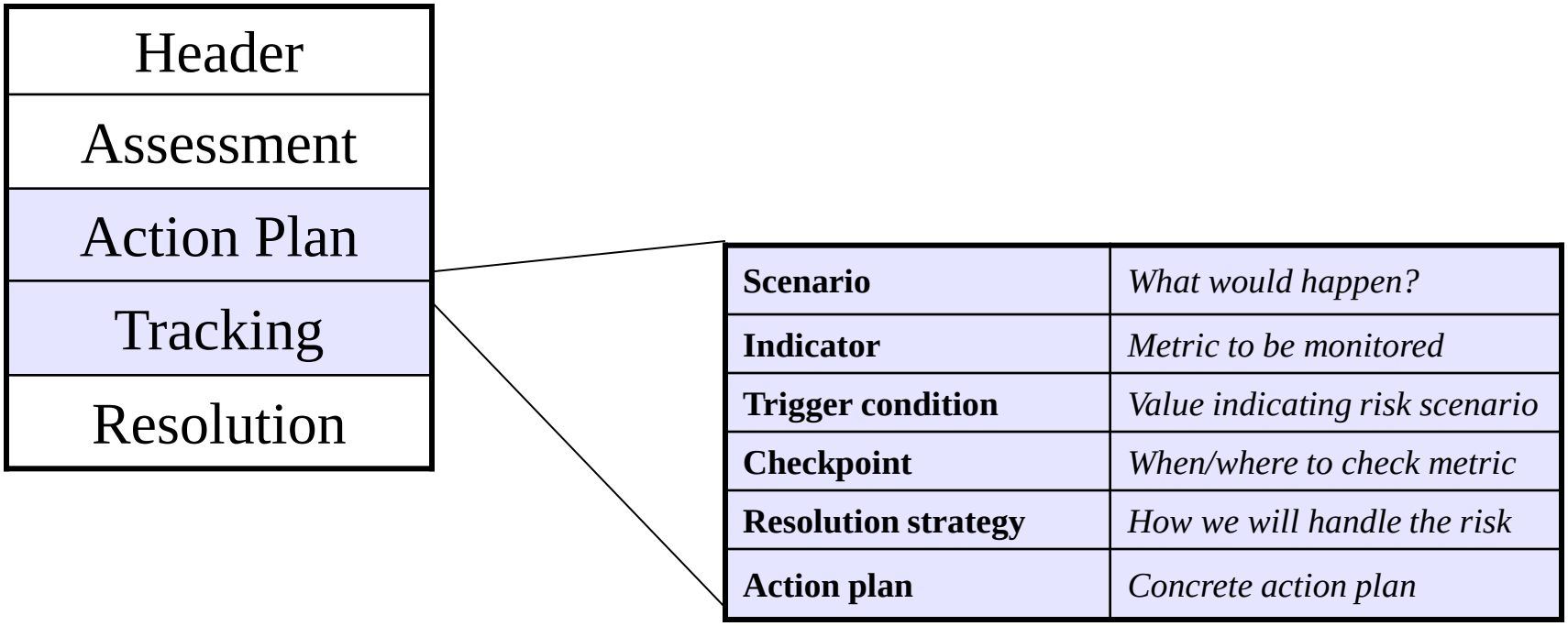
Analysis: Documentation

Adapted from *Managing Risk: Methods for Software Systems Development* by Elaine M. Hall, Addison-Wesley 1998



Planning/Tracking: Documentation

Adapted from *Managing Risk: Methods for Software Systems Development* by Elaine M. Hall, Addison-Wesley 1998



Resolution: Documentation

Adapted from *Managing Risk: Methods for Software Systems Development* by Elaine M. Hall, Addison-Wesley 1998