Risk Management

Where are we?

- 1. Introduction
- 2. Project Life Cycles
- 3. Project Artifacts
- 4. Work Elements, Schedule, Budget
- 5. Risk Management
  - Risk Management Plan

- Optional Inclusions

Outline

- What is Risk?
- Risk Identification: Threshold of Success
- Risk Management Plan
- Monitoring and Mitigation of Risks

Outcomes

- Understand the key parts of the Risk Management Plan
- Know how to identify the key parts of the Threshold of Success (TOS) for a project.
- Be able to write a TOS and begin writing Risks that can impact the TOS.
- Have a clear understanding of what it means to mitigate a Risk.

What is Risk?

A risk is a potential future harm that may arise from some present action

(-> *all risks have some probability rate of occurrence*)

What is Risk?

- There are a very few projects with no risk
- Software projects are fraught with cost overrun and schedule delays
- Risk management is an integral part of software project management

What is Risk?

- Risk is the result of making decisions
- Every decision has two outputs, a solutions and some risk
- Risk is neither good or bad it just is, the impact might be good or bad
- Risk has a probability and an impact

  “A problem is a risk whose time has come”

- All Risks should be based on a fact
- Examples:
  - Good: A condition exists therefore this event might occur…
  - Bad: A condition might exist therefore …

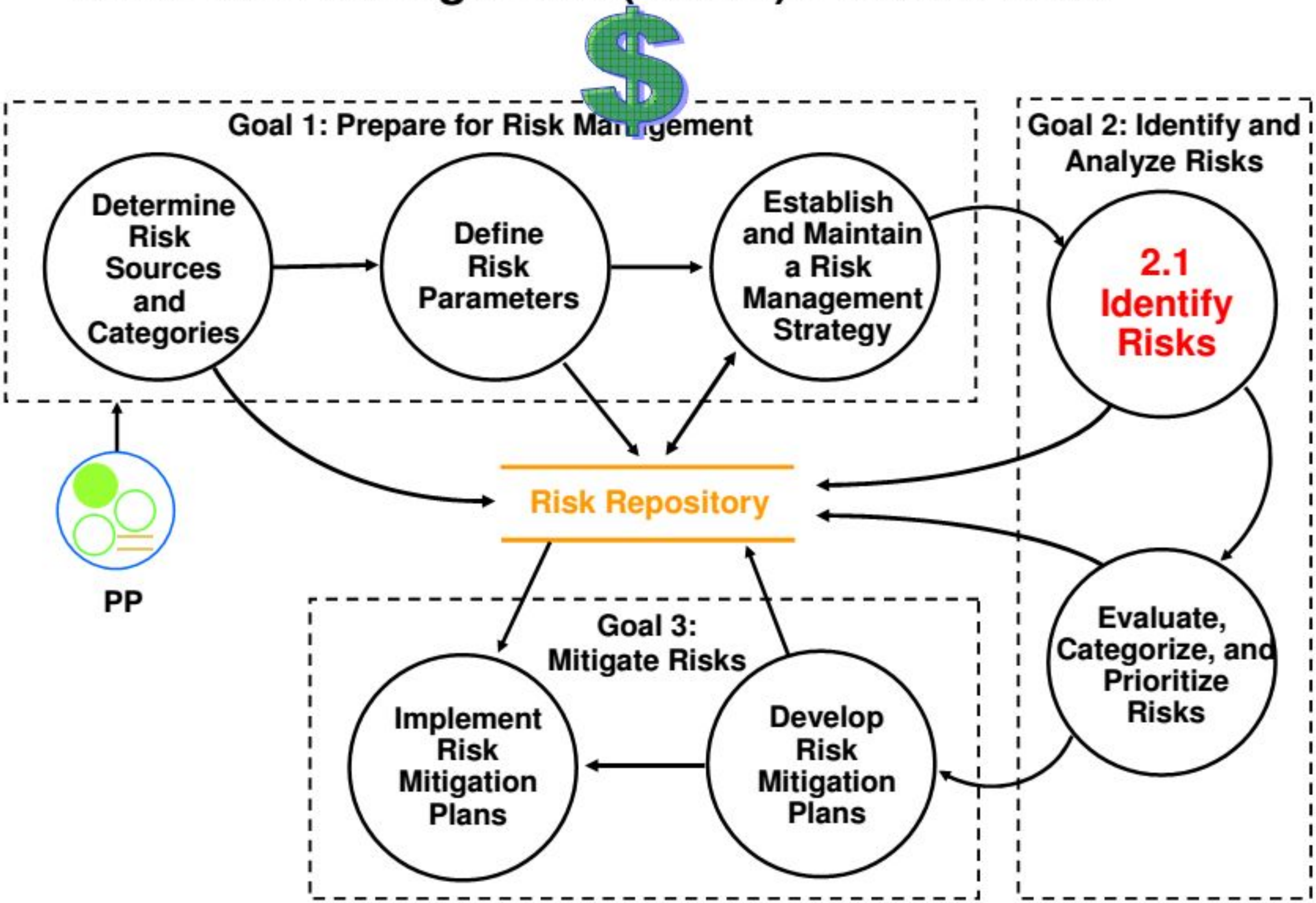
Key of Risk Management

- Risk Identification
- Risk Prioritization
- Risk Mitigation

Also:

- Risk Analysis – Take some time to consider risk
- Risk Monitoring – Have a method to identify the status of risks as they change

CMMI Risk Management (RSKM) Process Area


Goal 1: Prepare for Risk Management
Determine Risk Sources and Categories
Define Risk Parameters
Establish and Maintain a Risk Management Strategy
Goal 2: Identify and Analyze Risks
2.1 Identify Risks
Evaluate, Categorize, and Prioritize Risks
Risk Repository
PP
Goal 3: Mitigate Risks
Implement Risk Mitigation Plans
Develop Risk Mitigation Plans

Adult games

- *A team is put together* ***to build some software****. Neither the clients nor the team talk about the objectives of the project other than building "some software". After a few months,* ***something goes wrong*** *or* ***someone doesn't like*** *what's happening so* ***someone changes the rules****. Before too long, one side or the other is upset that they can't win, somebody throws a fit, and goes home. Instead of summoning invisible armor, software projects change the rules by cutting features, adding more requirements, moving due dates, wasting resources, and things like that.*

Threshold of Success

Defining and committing to a clear picture of success establishes the common ground rules for a project by making the basic project goals explicit. The technique is known as Threshold of Success.

Threshold of Success

- Clearly identifies what the project must minimally do to make the customer satisfied

- Establishes what are “must have” things versus “nice to have” items for the project

- Provides a clear view of what must be done and therefore a clear view of what might impact what must be done
  - i.e., The risks of the project.

Threshold of Success

A good Threshold of Success is made up of about 3-4 SMART goals (no more than a few bullets on a single PowerPoint slide).

SMART is a mnemonic which stands for -

Short/Specific

Measurable

Achievable

Relevant

Time bound.

Threshold of Success

Building a Threshold of Success

The easiest way to create a Threshold of Success is to first create a minimum picture of failure, then convert failure into success.

Example:

Failure for my current project might look something like this.

- Essential features are not ready by the end of the second quarter.
- Team members are dissatisfied or bored with their jobs.
- Newly hired team members don't feel like they're part of the team by March 31.
- There isn't enough money to continue development after this fiscal year and we have to fire people.

Threshold of Success

The threshold of success for my current project might look something like this.

- By the end of the second quarter, all "Must Have" features are implemented and pass acceptance tests with no known critical defects.
- All team members give average score of 5 or better on a job satisfaction survey taken quarterly.
- By March 31, the team has successfully executed at least three team building activities with all team members present.
- Funds of at least $1 million are secured by December 31 to allow for future development without a reduction in team size.

Threshold of Success

ToS statement:s

We MUST do X or have shown that our product has met at least Y to reach our ToS.

Risk management plan

As part of a larger, comprehensive project plan, the risk management plan outlines the response that will be taken for each risk—if it materializes

Risk management plan

Five main risk impact areas in SD:

- New, unproven technologies
- User and functional requirements
- Application and system architecture
- Performance
- Organizational

Risk management plan

- **New, unproven technologies.** The majority of software projects entail the use of new technologies. Training and knowledge are of critical importance, and the improper use of new technology most often leads directly to project failure.

Risk management plan

- **User and functional requirements.** Software requirements capture all user needs with respect to the software system features, functions, and quality of service. Change in elemental requirements will likely propagate throughout the entire project, and modifications to user requirements might not translate to functional requirements.

Risk management plan

- **Application and system architecture.** Taking the wrong direction with a platform, component, or architecture can have disastrous consequences. As with the technological risks, it is vital that the team includes experts who understand the architecture and have the capability to make sound design choices.

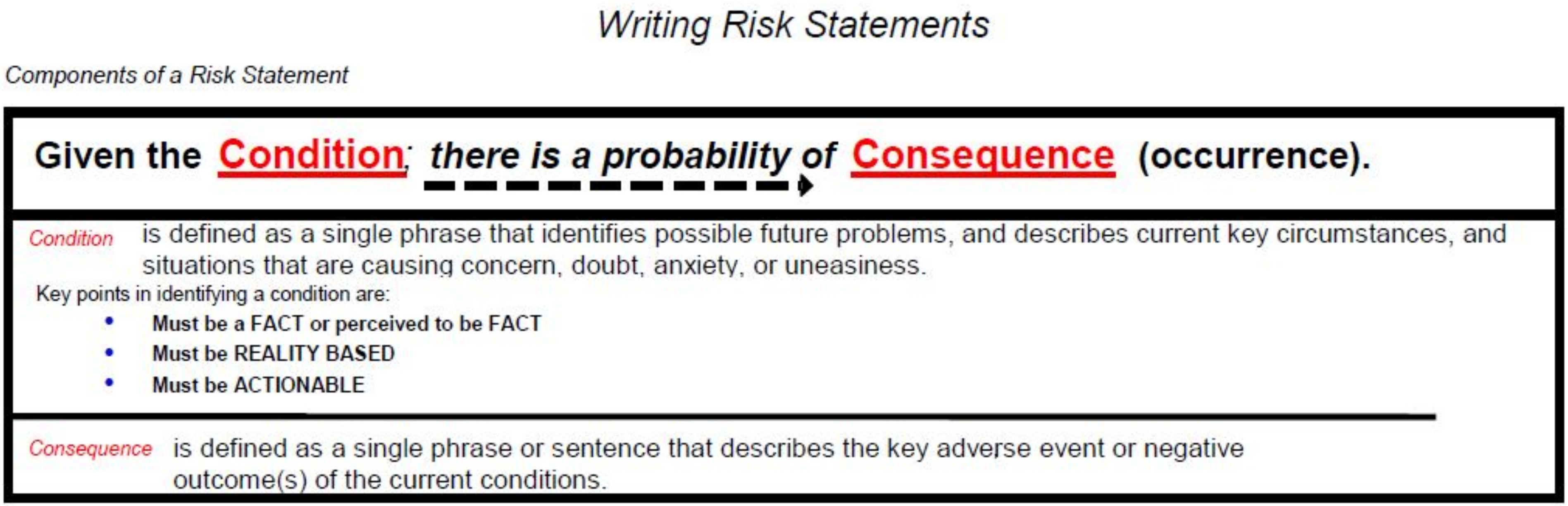
Risk management plan

- **Performance**. It’s important to ensure that any risk management plan encompasses user and partner expectations on performance. Consideration must be given to benchmarks and threshold testing throughout the project to ensure that the work products are moving in the right direction.

Risk management plan

- **Organizational**. Organizational problems may have adverse effects on project outcomes. Project management must plan for efficient execution of the project, and find a balance between the needs of the development team and the expectations of the customers.

Writing Risk Statements



Writing Risk Statements

Components of a Risk Statement

Given the Condition; *there is a probability of* Consequence (occurrence).

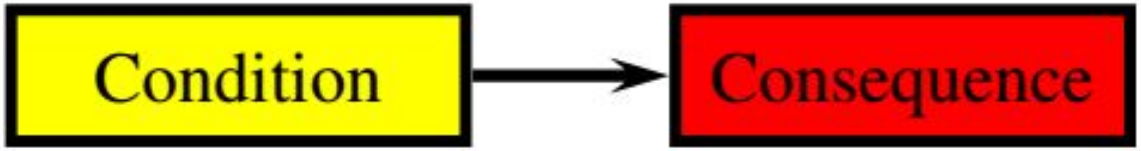
Condition is defined as a single phrase that identifies possible future problems, and describes current key circumstances, and situations that are causing concern, doubt, anxiety, or uneasiness.

Key points in identifying a condition are:

- **Must be a FACT or perceived to be FACT**
- **Must be REALITY BASED**
- **Must be ACTIONABLE**

Consequence is defined as a single phrase or sentence that describes the key adverse event or negative outcome(s) of the current conditions.

Writing Risk Statements

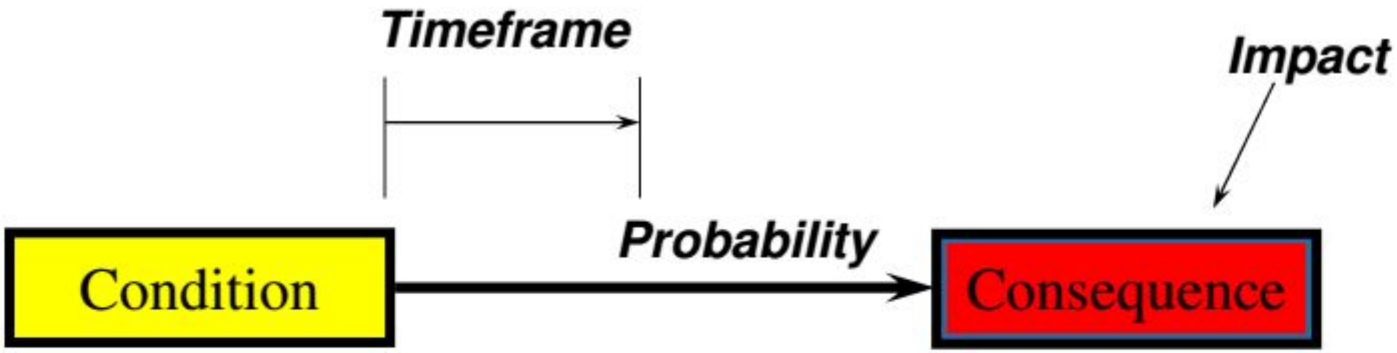


Condition

Consequence

The requirements for passing Milestone B have not yet been defined ; the current schedule and cost estimates may be inadequate.

Writing Risk Statements

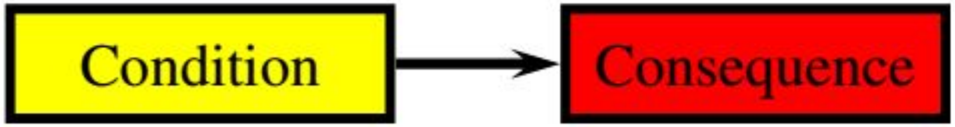


The requirements for passing Milestone B have not yet been defined ; the current schedule and cost estimates may be inadequate.

Writing Risk Statements

A “standard” format for risk statements provides:

- clarity
- consistency
- a basis for future risk processing



Condition

Consequence

A good Risk Statement is

- fact-based
- actionable
- brief

Example

- Lack of executive sponsorship (maybe because of change in the Administration); time delays, frustrations, credibility, and morale, and [a department cosponsoring the project] may pull out of [the project].
- The majority of software-to-software interfaces are not defined & controlled; incomplete interfaces results in no benefits from [the project].
- There has been inadequate schedule discipline (milestones, slippage, monitor progress, good project management) on this project; with no intervention the project will continue to slip & slide.

Table 1: General Categories of Risk

<table>
<tr><th colspan="2">Generic Risks</th><th colspan="2">Product-Specific Risks</th></tr>
<tr><th>Project Risks</th><th colspan="2">Product Risks</th><th>Business Risks</th></tr>
<tr><td colspan="4">Factors to consider:
People, size, process, technology, tools, organizational, managerial, customer, estimation, sales, support</td></tr>
</table>

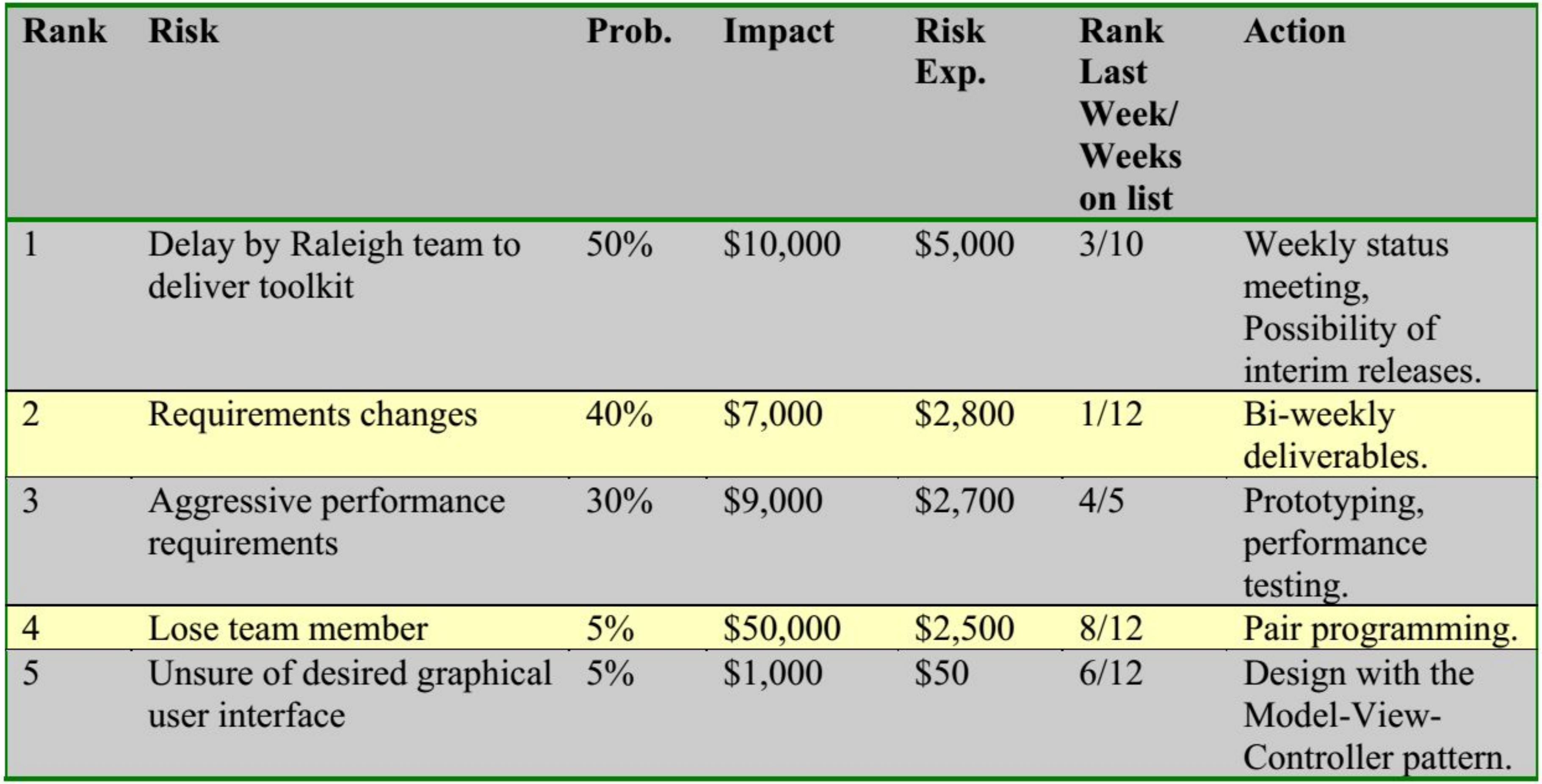
Risk prioritization

- Probability
  - Very Improbable
  - Improbable
  - Probable
  - Frequent
- Impact
  - Negligible
  - Marginal
  - Critical
  - Catastrophic
- Numerical Value

  Risk Exposure (RE)= P * C

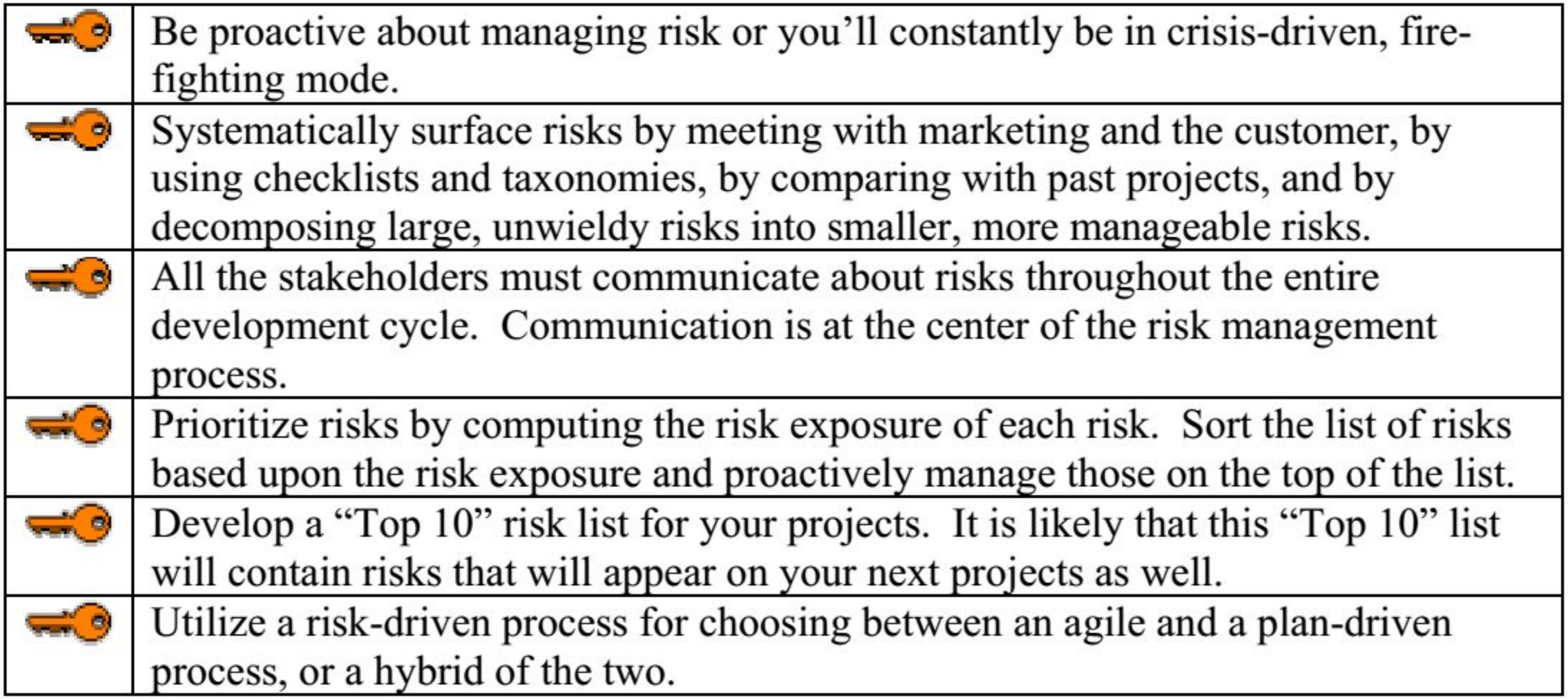
Risk Item	Risk Management Technique
Personnel shortfall	Staffing with top talent, job matching, team building, key personnel agreements, cross training
Unrealistic schedules and budgets	Detailed milestone cost and schedule estimation, design to cost, incremental development, software reuse, requirements scrubbing
Developing the wrong functions and properties	Organizational analysis, mission analysis, operations-concept formulation, user surveys and user participation, prototyping, early users' manuals
Developing the wrong user interface	Prototyping, scenarios, task analysis, user participation
Gold-plating (e.g. implementing "neat features" not asked for by customer)	Requirements scrubbing, prototyping, cost-benefit analysis, designing to cost
Continuing stream of requirements changes	High change threshold information hiding, incremental development (deferring changes to later increments)
Shortfalls in externally-furnished components (e.g. component reuse)	Benchmarking, inspections, reference checking, compatibility analysis
Shortfalls in externally performed tasks (e.g. worked performed by a contractor)	Reference checking, pre-award audits, award-fee contracts, competitive design or prototyping, team building
Real-time performance shortfalls	Simulation, benchmarking, modeling, prototyping, instrumentation, tuning
Straining computer science capabilities	Technical analysis, cost-benefit analysis, prototyping, reference checking

Table of risks



Rank	Risk	Prob.	Impact	Risk Exp.	Rank Last Week/ Weeks on list	Action
1	Delay by Raleigh team to deliver toolkit	50%	$10,000	$5,000	3/10	Weekly status meeting, Possibility of interim releases.
2	Requirements changes	40%	$7,000	$2,800	1/12	Bi-weekly deliverables.
3	Aggressive performance requirements	30%	$9,000	$2,700	4/5	Prototyping, performance testing.
4	Lose team member	5%	$50,000	$2,500	8/12	Pair programming.
5	Unsure of desired graphical user interface	5%	$1,000	$50	6/12	Design with the Model-View-Controller pattern.

Key Ideas for Risk Management:



🔑	Be proactive about managing risk or you'll constantly be in crisis-driven, fire-fighting mode.
🔑	Systematically surface risks by meeting with marketing and the customer, by using checklists and taxonomies, by comparing with past projects, and by decomposing large, unwieldy risks into smaller, more manageable risks.
🔑	All the stakeholders must communicate about risks throughout the entire development cycle. Communication is at the center of the risk management process.
🔑	Prioritize risks by computing the risk exposure of each risk. Sort the list of risks based upon the risk exposure and proactively manage those on the top of the list.
🔑	Develop a "Top 10" risk list for your projects. It is likely that this "Top 10" list will contain risks that will appear on your next projects as well.
🔑	Utilize a risk-driven process for choosing between an agile and a plan-driven process, or a hybrid of the two.

Risk mitigation

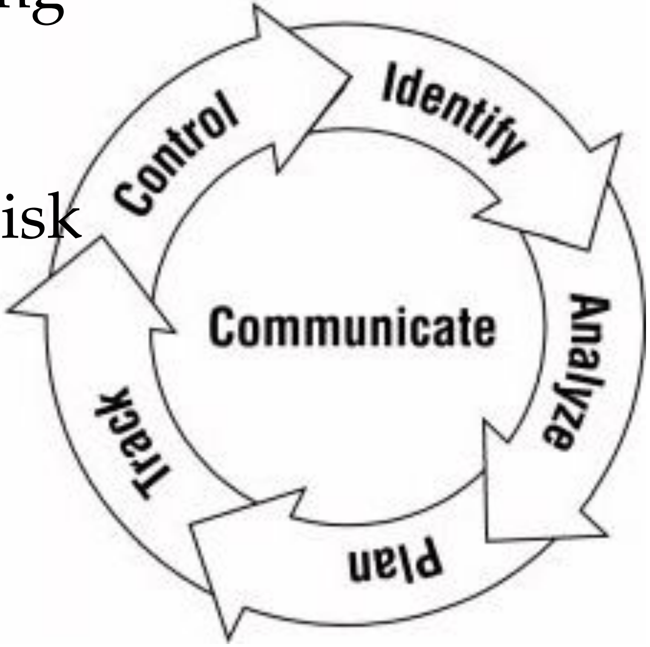
Risk management includes the following tasks:

- ***Identify*** risks and their triggers
- ***Classify*** and **prioritize** all risks
- Craft a ***plan*** that links each risk to a mitigation
- ***Monitor*** for risk triggers during the project
- Implement the ***mitigating*** *action* if any risk materializes
- ***Communicate*** risk status throughout project

Risk mitigation

Evaluation Project Decisions gives these activities

- Defining a Threshold of Success
- Identifying risks
- Formulating risk statements
- Mitigating, tracking and controlling risk
- Communicating about risk
- Trading off resources to manage risk



Summary:

- A work team identifying risks needs to agree on an end-point against which to identify and analyze the risks.
- There needs to be a standard way of capturing (documenting) a risk.
- Facilitators need practice to become comfortable writing risks in front of a group.
- There are many ways for program management to support good risk identification:
  - Encourage documentation of risks privately at the working team level
  - Integrate risk identification and management into normal project management
  - Accept any risk identified into the repository – don’t “vet them out”
  - Acknowledge that the program’s decision-makers are the real “risk managers,” and have the decision-makers step up to the job