Software Risk Management

Lecturer: Ngo Huy Bien
Software Engineering Department
Faculty of Information Technology
VNUHCM - University of Science
Ho Chi Minh City, Vietnam
nhbien@fit.hcmus.edu.vn

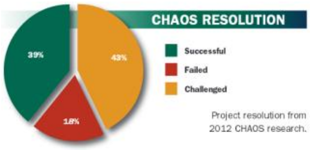
Objectives

- To *identify* project risks
- To *analyze* project risks
- To propose risk *response*
- To create risk management *plan*
- To *monitor* and *track* project risks

References

1. C. Ravindranath Pandian. Applied Software Risk Management A Guide for Software Project Managers. 2007.
2. BW Boehm. Software risk management: principles and practices. 1991.
3. M. T. Taghavifard et al. Decision Making Under Uncertain and Risky Situations. 2009.
4. RD Shachter. Evaluating Influence Diagrams. 1986.
5. Daniel D. Galorath. Software Sizing, Estimation, and Risk Management. 2006.
6. Hooman Hoodat and Hassan Rashidi. Classification and Analysis of Risks in Software Engineering. 2009.
7. Kathy Schwalbe. An Introduction to Project Management. Fifth Edition. 2015.
8. Glen J. Briscoe. Risk Management Guide. 1982.

Why Projects Failed?



Project resolution from 2012 CHAOS research.

2012 CHAOS Report
standishgroup.com

planned scope, resources, cost and schedule

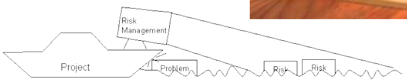
final project cost, performance, and schedule

known unknowns (future events)

All software projects are unique undertakings.

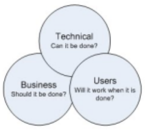
Risk [1]

- *Risk* is the possibility of something bad happening at some time in the future.
- *Risk* is the probability of suffering loss while pursuing goals due to factors that are unpredictable or beyond.



Risk Identification

Risk Identification produces lists of the project-specific risk items likely to compromise a project's satisfactory outcome.



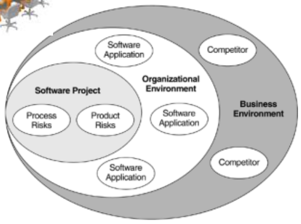
- Technical risks
- Management risks
- Financial risks
- Contractual and legal risks
- User acceptance risks
- Maintainability risks

How to Indentify Risks?

1. Meeting

2 *Brainstorming*

- *Risk taxonomies*
- Decomposition
- Critical path analysis
- Interviewing
- Assumption analysis
- Voluntary reporting



Past projects comparison

Software Risk Check List (I)

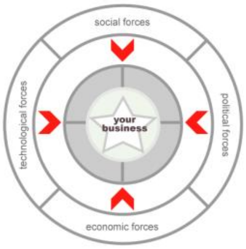
- Lack of Leadership
- Unclear requirements
- Too many requirement changes
- Unrealistic schedules
- Working on new technology
- Working on new process
- Manpower attrition
- Slow performance

Software Risk Check List (II)

- The Mid-Course Correction
- Padded Estimates Generate Distrust
- Self-Fulfilling Prophecy
- Working Backward From a Deadline
- Misunderstood Predecessors
- Iteration Abuse
- Problems Are Found Too Late
- Big, Useless Meetings
- The Indispensable "Hero"

PEST Analysis

PEST analysis is a strategic planning method used to review a strategy or position, direction of a company, a marketing proposition, or idea in a project.



- *Brainstorm the relevant factors* that apply to you
- *Identify the information* that applies to these factors
- *Draw conclusions* from this information

SWOT Analysis

SWOT analysis is a strategic planning method used to evaluate the Strengths, Weaknesses, Opportunities, and Threats involved in a project.

Strengths

- Skills, prices, motivation, resources

Weaknesses

- Experiences, skills, time

Objective

Opportunities

- Learning, skill and knowledge improvement

Threats

- Competitors, changes, team work

Risk Identification Product

TABLE I
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	

TABLE II
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	

Risk Analysis

Risk analysis is the scientific measurement of the degree of danger or hazard involved in any operation or activity.

– "*How risky* are my risks?"

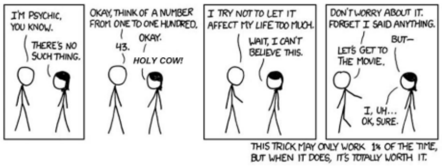
Risk Characterization

Probability	Description
0% – 20%	Remote
20% – 40%	Unlikely
40% – 70%	Likely
70% – 90%	Highly likely
90% – 100%	Nearly certain

Time of Occurrence
In next month
One to two months from now
Three or more months from now

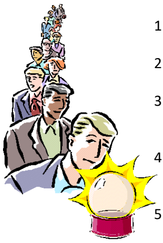
Impact (Slips to schedule and budget)	Description
< 10% of total budget	Minimal or no impact
< 15% of total budget	Small, acceptable
< 50% of total budget	Moderate system damage
< 60% of total budget	Significant impact to project
>= 60% of total budget	Failure of project operations

Uncertainty [7]


I'M PSYCHIC, YOU KNOW.
THERE'S NO SUCH THING.
OKAY, THINK OF A NUMBER FROM ONE TO ONE HUNDRED.
43.
OKAY.
HOLY COW!
I TRY NOT TO LET IT AFFECT MY LIFE TOO MUCH.
WAIT, I CAN'T BELIEVE THIS.
DON'T WORRY ABOUT IT. FORGET I SAID ANYTHING.
BUT—
LET'S GET TO THE MOVIE.
I, UH... OK, SURE.
THIS TRICK MAY ONLY WORK 1% OF THE TIME, BUT WHEN IT DOES, IT'S TOTALLY WORTH IT.

Delphi Method

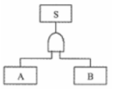
Delphi method is a method for structuring a group communication process so that the process is effective in allowing a group of individuals, as a whole, to deal with a complex problem.



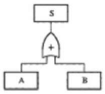
1. Develop initial questionnaire and distribute it to the panel.
2. Panelists independently generate their ideas in answer to the questionnaire and return it.
3. The moderator summarizes the responses to the first questionnaire and develops a feedback report along with the second set of questionnaires for the panelists.
4. Having received the feedback report, panelists independently evaluate earlier responses and independently vote on the second questionnaire.
5. The moderator develops a final summary and feedback report to the group and decision makers.

Risk Tree Analysis [6]

- *Risk tree* is a *graphical model* of various combinations of risks that result in the occurrence of the predefined undesired event.
- To analyze *using* risk tree, it is necessary to
  - specify the *undesired state* of the system. This state may be the failure of the system or of a subsystem.
  - Then a list is made of all the *possible ways* in which these events can occur.
  - Each of the possible ways is then examined independently to find out how it can occur.

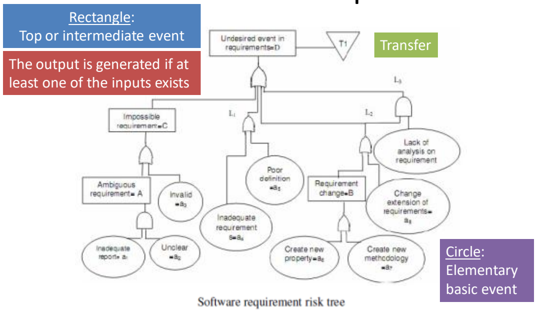


AND Gate
P(S) =P (A).P (B)



OR Gate
P(S) =1-(1-P (A)) (1-P (B))

Risk Tree Example


Rectangle:
Top or intermediate event
The output is generated if at least one of the inputs exists
Undesired event in requirements=D
T1
Transfer
L₃
L₁
L₂
Impossible requirement=C
Lack of analysis on requirement
Ambiguous requirement= A
Invalid =a₃
Poor definition =a₅
Requirement change=B
Change extension of requirements= a₈
Inadequate requirement s=a₄
Inadequate report= a₁
Unclear =a₂
Create new property=a₆
Create new methodology =a₇
Circle:
Elementary basic event

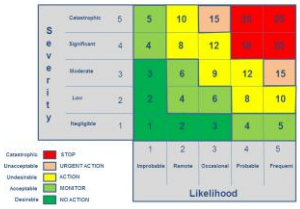
Software requirement risk tree

Risk Exposure (Priority)

- Risk exposure is defined as

  Risk Exposure (Priority) = Probability * Impact.

- It is simply the <u>average</u> *impact*.



Severity		1	2	3	4	5
Catastrophic	5	5	10	15	20	25
Significant	4	4	8	12	16	20
Moderate	3	3	6	9	12	15
Low	2	2	4	6	8	10
Negligible	1	1	2	3	4	5
		Improbable	Remote	Occasional	Probable	Frequent

Likelihood

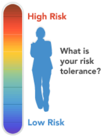
Catastrophic	🟥	STOP
Unacceptable	🟧	URGENT ACTION
Undesirable	🟨	ACTION
Acceptable	🟩	MONITOR
Desirable	🟩	NO ACTION

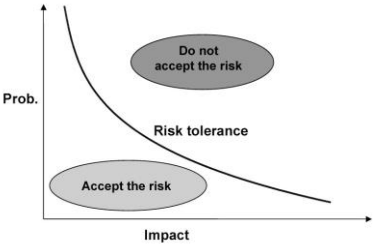
Risk Analysis Product [2]

Unsatisfactory Outcome (UO)	Prob(UO)	Loss(UO)	Risk Exposure
A. S/W error kills experiment	3-5	10	30-50
B. S/W error loses key data	3-5	8	24-40
C. Fault tolerance features cause unacceptable performance	4-8	7	28-56
D. Monitoring software reports unsafe condition as safe	5	9	45
E. Monitoring software reports safe condition as unsafe	5	3	15
F. Hardware delay causes schedule overrun	6	4	24
G. Data reduction software errors cause extra work	8	1	8
H. Poor user interface causes inefficient operation	6	5	30
I. Processor memory insufficient	1	7	7

Risk Tolerance

Risk tolerance is the willingness of some person or some organization to <u>accept</u> or <u>avoid</u> risk.





Risk Response

- *Acceptance*: To acknowledge the risk's existence, but to take no preemptive action to resolve it, except for the possible development of <u>contingency plans</u> should the risk event come to pass.
- *Avoidance*: To eliminate the conditions that allow the risk to be present at all, most frequently by dropping the project or the task.
- *Deflection*: To transfer the risk (in whole or part) to another organization, individual, or entity.
- *Mitigation*: To minimize the probability of a risk's occurrence or the impact of the risk should it occur.

Example of Risk Mitigation (I)

- *Inexperience* with project process/technology
  - Make estimates with a buffer for initial learning time
  - Maintain buffers of extra resources
  - Define a project-specific training program
  - Conduct cross-training sections
- *Unclear* requirements and/or acceptance criteria
  - Use experience and logic to make some assumptions and keep the client informed; obtain sign-off
  - Develop a prototype and have the requirements reviewed by the client.

Example of Risk Mitigation (II)

- Too many *requirement changes*
  - Obtain sign-off for the initial requirements specification from the customer.
  - Convince the customer that changes in requirements will affect the schedule
    - If the requirements are changing rapidly we should RE-ESTIMATE the final delivery date and NOTIFY the manger about the new schedule AS SOON AS we receive a change request.
    - We should NOT let the manager know the effect on the last day.
  - Define a procedure to handle requirement changes
  - Negotiate payment on actual effort
- *Unrealistic schedules*
  - Negotiate for a better schedule.
  - Identify parallel tasks.
  - Have resources ready early
  - Identify areas that can be automated, COTS components
  - If the critical path is not within the schedule, negotiate with the client
  - Negotiate payment on actual effort.

Example of Risk Mitigation (III)

- Team spirit and *attitude*
  - Ensure that multiple resources are assigned on key project areas.
  - Have team-building sessions.
  - Rotate jobs among team members
  - Keep extra resources in the project as backup.
  - Maintain proper documentation of each individual's work.
  - Follow the configuration management process and guidelines strictly.
- Not meeting *performance requirements*
  - Define the performance criteria clearly and have them reviewed by the client.
  - Define standards to be followed to meet the performance criteria
  - Prepare the design to meet performance criteria and review it.
  - Simulate or prototype performance of critical transactions.
  - Test with a representative volume of data where possible.
  - Conduct stress tests where possible.

Example of Risk Mitigation (IV) [2]

1	Personnel shortfalls	Staffing with top talent; job matching (matching the right people with the right job); teambuilding; morale building; cross-training ; prescheduling key people
2	Unrealistic schedules and budgets	Detailed multisource costs & schedule estimation; design to cost; incremental development; software reuse; requirements scrubbing
3	Developing the wrong software functions	Organization analysis; mission analysis; ops-concept formulation ; user survey; prototyping; early users' manuals
4	Developing the wrong user interfacing	Prototyping; scenarios; task analysis
5	Gold plating	Requirements scrubbing; prototyping; cost-benefit analysis; design to cost
6	Continuing stream of requirements changes	High change threshold; information hiding; incremental development (defer changes to later increments)
7	Shortfalls in externally furnished components	Benchmarking; inspections; reference checking; compatibility analysis
8	Shortfalls in externally performed tasks	Reference checking; pre-award audits; award-fee contacts; competitive design or prototyping; teambuilding
9	Real-time performance shortfalls	Simulation; benchmarking; modeling; prototyping; instrumentation; tuning
10	Straining computer science capabilities	Technical analysis; cost-benefit analysis; prototyping; reference checking.

Buying Information

- Scenarios
- Modeling
- Prototyping
- Task analysis
- Software reuse
- Cost-Benefit analysis

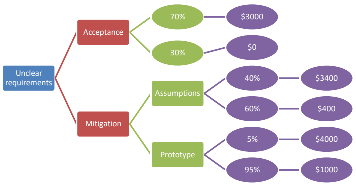
Some risk is necessary for potential reward. You can avoid risks by investing your money in a bank savings account, but you won't get rich that way.

Which Response to Choose? [2]

- *Unclear* requirements **(70%**, **$3000**)
  - Use experience and logic to make some assumptions and keep the client informed (**40%**, **$3000**, **$400**).
  - Develop a prototype and have the requirements reviewed by the client (**5%**, **$3000**, **$1000**).

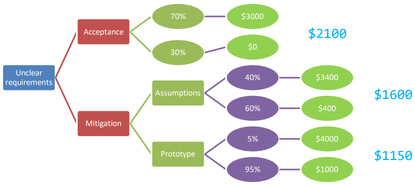
Decision Tree [3]

- *A decision tree* is a chronological representation of the decision process.
- It utilizes a network of two types of nodes:
  - *decision (choice) nodes* (represented by square shapes), and
  - states of *nature (chance) nodes* (represented by circles).



How to *Use* a Decision Tree?

- Draw the decision tree using squares to represent decisions and circles to represent uncertainty,
- Evaluate the decision tree to make sure all possible outcomes are included,
- Calculate the *tree values* working from the right side back to the left,
- Calculate the values of uncertain outcome nodes by multiplying the value of the outcomes by their probability (i.e., *expected values* or average *impact*).

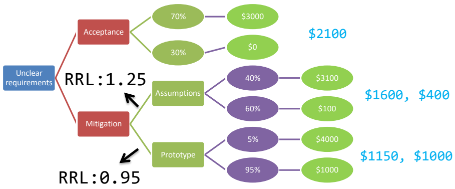


Risk Reduction Leverage [2]

Risk reduction leverage provides a cost benefit analysis of treatment options. (*1$ reduces ? average impact*)

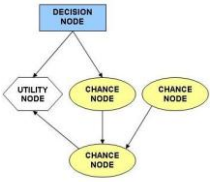
Risk Reduction Leverage =

$(RE_{before} – RE_{after})$ / Risk Reduction Cost



Influence Diagram [4]

- An *influence diagram* is a graphical structure for modeling uncertain variables and decisions and explicitly revealing probabilistic dependence and the flow of information.
- Influence diagram consists of nodes or variables connected by directed arrows. There are three kinds of nodes:
  - (1) *decision nodes* representing alternative actions that can be taken by decision makers;
  - (2) *chance nodes* representing events or system variables that are outcomes of the decision or other chance variables;
  - (3) *value or utility nodes*, variables that summarize the final outcome of a decision.



Risk Management Planning

Risk management planning produces plans for addressing each risk item (e.g., via risk avoidance, risk transfer, risk reduction, or buying information), including the coordination of the individual risk-item plans with each other and with the overall project plan.

- The plan is organized around a standard format for software plans, oriented around answering the standard questions of "*why, what, when, who, where, how, and how much*."
- This plan organization allows the plans to be *concise* (e.g., fitting on one page), *action-oriented*, easy to understand, and easy to monitor.

[IEEE Std 1540-2001]

Risk Management Plan Example (I)

<u>Risk Management Plan: Fault Tolerance Prototyping</u> **1 Risk**

1. Objectives (The "Why")

- Determine, reduce level of risk of the software fault tolerance features causing unacceptable performance
- Create a description of and a development plan for a set of low-risk fault tolerance features

2. Deliverables and Milestones (The "What" and "When")

- By week 3
  1. Evaluation of fault tolerance option
  2. Assessment of reusable components
  3. Draft workload characterization
  4. Evaluation plan for prototype exercise
  5. Description of prototype
- By week 7
  6. Operational prototype with key fault tolerance features
  7. Workload simulation
  8. Instrumentation and data reduction capabilities
  9. Draft Description, plan for fault tolerance features

-By week 10

10. Evaluation and iteration of prototype
11. Revised description, plan for fault tolerance features

Risk Management Plan Example (II)

3. Responsibilities (The "Who" and "Where")

- System Engineer: G. Smith
  Tasks 1, 3, 4, 9, 11, support of tasks 5, 10
- Lead Programmer: C. Lee
  Tasks 5, 6, 7, 10 support of tasks 1, 3
- Programmer: J. Wilson
  Tasks 2, 8, support of tasks 5, 6, 7, 10

4. Approach (The "How")

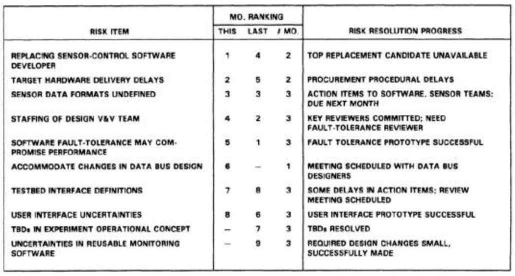
- Design-to-Schedule prototyping effort
- Driven by hypotheses about fault tolerance-performance effects
- Use real-time OS, add prototype fault tolerance features
- Evaluate performance with respect to representative workload
- Refine Prototype based on results observed

5. Resources (The "How Much")

$60K - Full-time system engineer, lead programmer, programmer (10weeks)*(3 staff)*($2K/staff-week)
$0K - 3 Dedicated workstations (from project pool)$0K - 2 Target processors (from project pool)
$0K - 1 Test co-processor (from project pool)
$10K – Contingencies
$70K - Total

Risk Monitoring

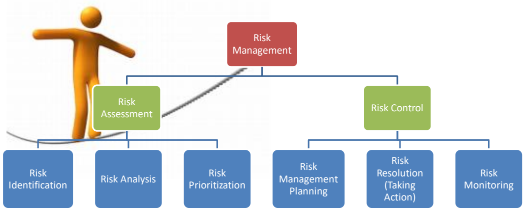
Risk monitoring involves tracking the project's progress towards resolving its risk items and taking corrective action where appropriate.



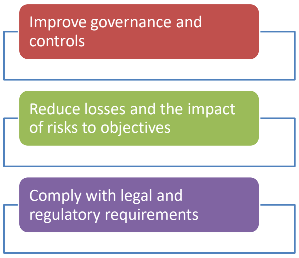
RISK ITEM	MO. RANKING			RISK RESOLUTION PROGRESS
	THIS	LAST	# MO.	
REPLACING SENSOR-CONTROL SOFTWARE DEVELOPER	1	4	2	TOP REPLACEMENT CANDIDATE UNAVAILABLE
TARGET HARDWARE DELIVERY DELAYS	2	5	2	PROCUREMENT PROCEDURAL DELAYS
SENSOR DATA FORMATS UNDEFINED	3	3	3	ACTION ITEMS TO SOFTWARE, SENSOR TEAMS; DUE NEXT MONTH
STAFFING OF DESIGN V&V TEAM	4	2	3	KEY REVIEWERS COMMITTED; NEED FAULT-TOLERANCE REVIEWER
SOFTWARE FAULT-TOLERANCE MAY COMPROMISE PERFORMANCE	5	1	3	FAULT TOLERANCE PROTOTYPE SUCCESSFUL
ACCOMMODATE CHANGES IN DATA BUS DESIGN	6	–	1	MEETING SCHEDULED WITH DATA BUS DESIGNERS
TESTBED INTERFACE DEFINITIONS	7	8	3	SOME DELAYS IN ACTION ITEMS; REVIEW MEETING SCHEDULED
USER INTERFACE UNCERTAINTIES	8	6	3	USER INTERFACE PROTOTYPE SUCCESSFUL
TBDs IN EXPERIMENT OPERATIONAL CONCEPT	–	7	3	TBDs RESOLVED
UNCERTAINTIES IN REUSABLE MONITORING SOFTWARE	–	9	3	REQUIRED DESIGN CHANGES SMALL, SUCCESSFULLY MADE

Risk Management

Risk management is a systematic approach to reducing the harm due to risks, making the project less vulnerable and the product more robust.



Why Risk Management?


Improve governance and controls
Reduce losses and the impact of risks to objectives
Comply with legal and regulatory requirements

"If you don't actively attack the risks, they will actively attack you." -- *Tom Gilb*

Thank You for Your Time