ISYS90050 IT Project and Change Management

Tutorial 4

Requirements terminology

- Be consistent in use of terms (use glossary if you have to) in documentation
  - Use the term 'shall' for *mandatory* requirements; 'should' for *desired* requirements
  - The terms 'will' and 'shall' are contractually binding and held up in court!
  - Avoid 'must'

Shall:

- be present as a *requirement and be verifiable* = > the system shall have a log in screen that has a unique identifier/user name for the user.
- Shall is used to indicate a requirement that is contractually binding, meaning it must be implemented, and its implementation verified.

Should:

- Goals, *non-mandatory* provisions. 'Should' is used to indicate a goal which must be addressed by the design team but is not formally verified. Eg. Ease of use? Should be easy to use for whom; what's easy for you may not be easy for me?

Problems with natural language specifications

- Ambiguity
  - The readers and writers of the requirement must interpret the same words in the same way. NL is naturally ambiguous so this is very difficult.
- Over-flexibility
  - The same thing may be said in a number of different ways in the specification.
- Lack of modularisation
  - Natural language structures are inadequate to structure system requirements.

Task 1: Write good requirements

- [ ] Clients must be able to make a selection from a range of menus
- [ ] Clients shall be able to pay using a range of payment methods including debits cards, credit cards and PayPal. In addition, clients must be able to use multiple payment methods for a single order.
- [ ] The application shall allow a guest list to be maintained for each event. The guest list shall have the first name, last name and contact number of the guest.
- [ ] The application must allow employee rostering and scheduling conflicts to be managed.
- [ ] The application shall support a minimum of 5000 concurrent users and a maximum of 10000 concurrent users to access the system.
- [ ] The application must allow aggregation of overall quantity of supplies for events within a specified time frame.
- [ ] The application must support a messaging system to key personnel at event facilities.
- [ ] All the User Interfaces shall have the logo of Catherine's Catering Business (to be shared to the development team by Catherine).
- [ ] Feedback from clients must be recorded in the application.
- [ ] System must generate reports of each event or group of events, these reports must be printable.

Task 2: Write good requirements

- [ ] The application must be secure (because we don't want outsiders to know when and where particular events might be scheduled or who might be turning up to that event (guest lists).
- [ ] The application should be easy to use for all clients.
- [ ] The application must be reliable.
- [ ] The application must load very fast.

Task 3: Write good requirements

- [ ] The application shall present different menus to clientele. All the menus shall be shared with the development team by Catherine.
- [ ] The application must present a range of venues to clientele.
- [ ] Pricing of the items in the menu should be comparable to competitors in the market.
- [ ] Pictures of the menu items should be attractive.
- [ ] The website must clearly illustrate company policies and the refund policies.
- [ ] The application will allow the client to follow the Australian Privacy Act 1988.

What is a risk?

- A risk is a "hazard; peril; exposure to loss or injury"

Webster dictionary

- "[A risk is] an uncertain event or condition that, **if it occurs**, has a positive or negative effect on the project objectives"

PMBOK Definition

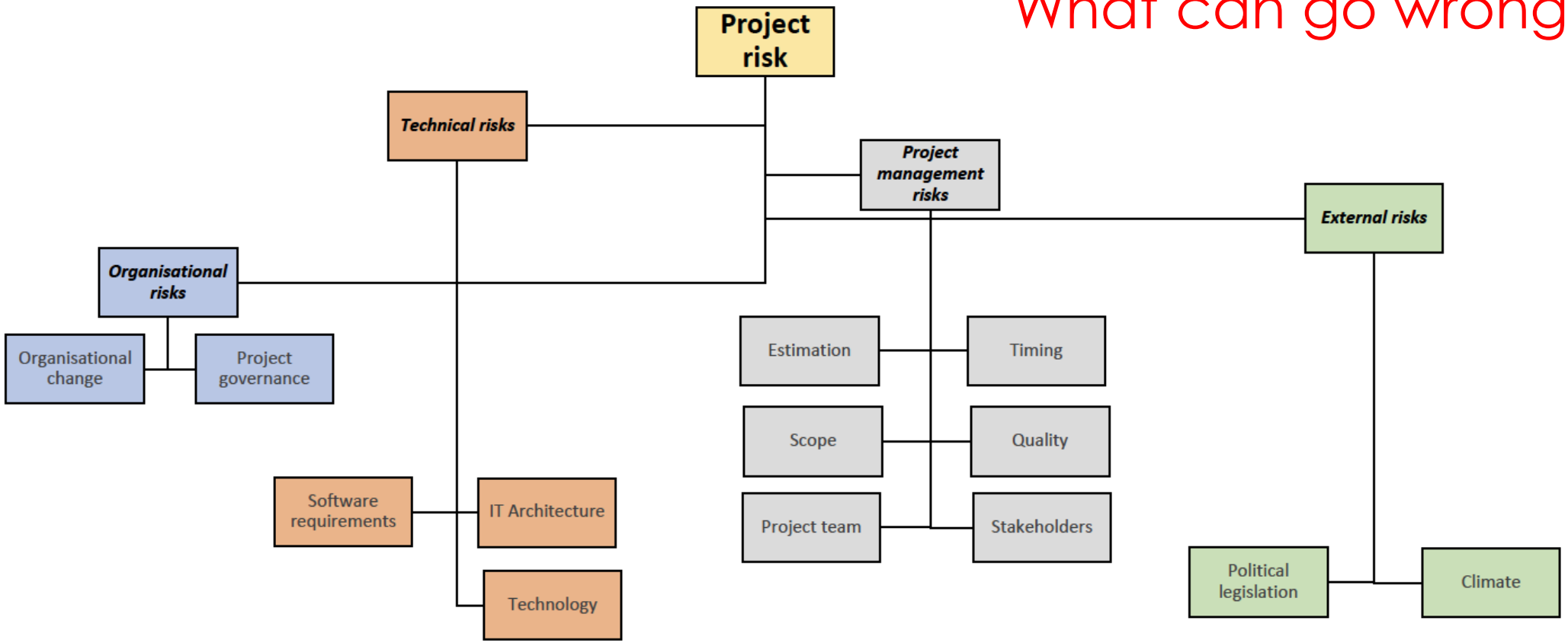
Immediate Impact

Consequence

Where to look for?

Risk breakdown structure/ framework (RBS)

What can go wrong?



Where to look for?

Risk Identification from Threat Categories

- Technical
- Cost
- Schedule
- Client
- Contractual
- Quality
- Financial
- Political
- Environmental
- People

What can go wrong?

How to write a risk?

Objectives

How to write a risk statement:

Event

- [Event that has an effect on objectives] **caused by** [cause/s] **resulting in** [consequence/s]

Immediate Impact

Example: <Schedule overrun by three weeks> caused by <tester resigning> resulting in <tasks related to testing units 3, 4, 5 will not be completed>

Event

(or)

Immediate Impact

- If <something happens- an action> then <what does it effect - consequences> thus leading to <impacts on the objectives>

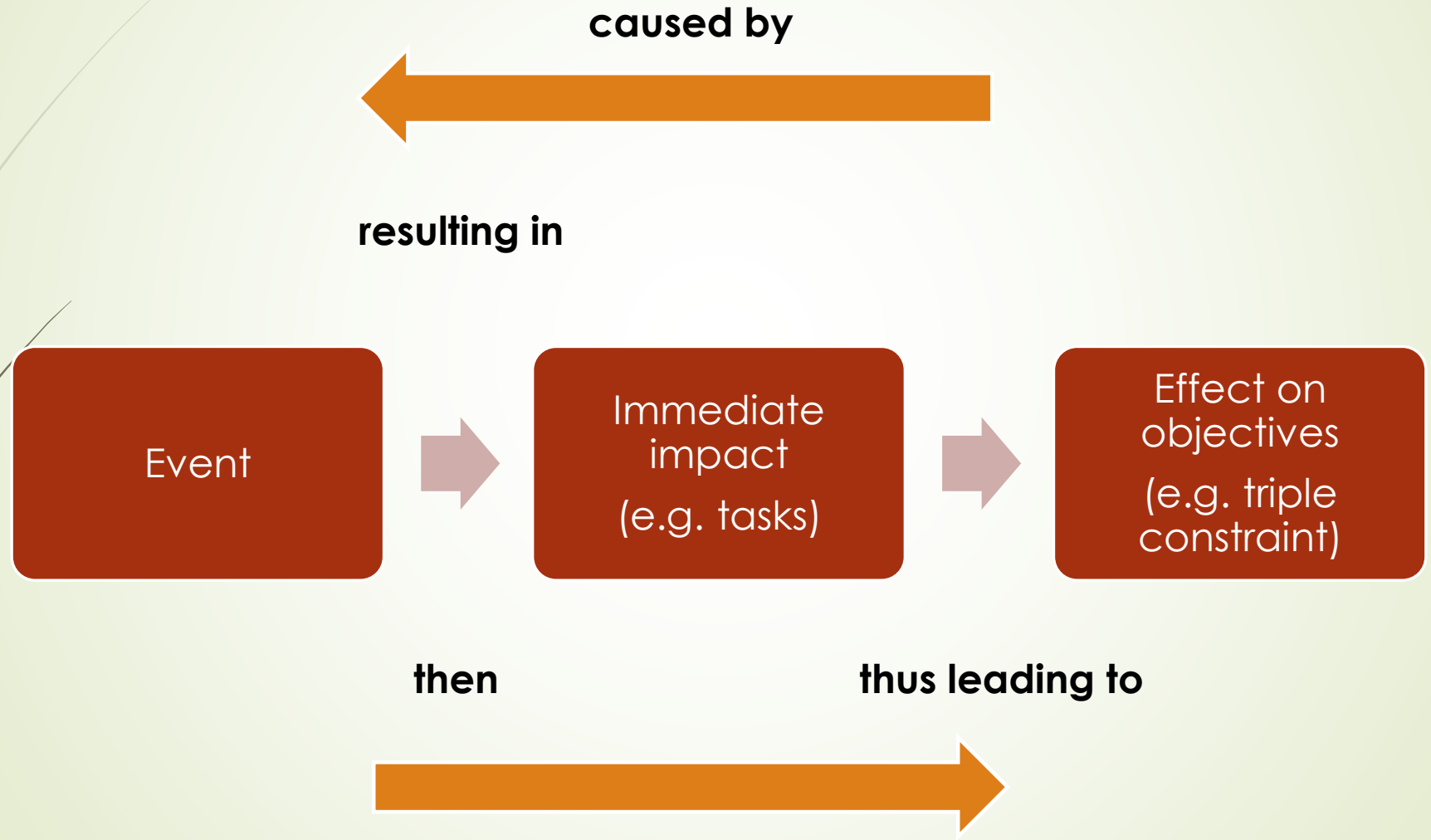
Action – consequence - impact

Objectives

Example: If <tester resigns> then <tasks related to testing units 3, 4, 5 will not be completed> thus leading to <schedule over-run by three weeks>

Note: you may use anyone of the above structures to write a risk.

How to write a risk?



How to identify risks?

There are several methods to conduct root cause analysis to identify risks:

Collect information through:

- Brainstorming/interviewing experts or experienced team members
- Nominal Group Technique

And represent using:

- Cause and effect diagram – fish bone method (OR)
- SWOT analysis (Strengths, Weaknesses, Opportunities and Threats

What could be a problem, what could be the reasons; what could it affect; how could it impact on the objectives of the project; when would it have affect (trigger);

Task 4: What are the risks and Why?

- What can happen?
- How and why, it can happen? Analyse possible causes and scenarios or description of the risk, incident or accident.

Task 5: What are the risks and Why?

- What is the probability (e.g. 20%) of it happening? Why?
- What will be the consequence if it happens?

Task 6: Risk Ranking

- How risks should be prioritized?

Risk Statement	Probability (0-99.9%)	Impact (0-10)	Exposure (P * I)	Risk Ranking
Requirement: The application shall allow employee rostering and scheduling conflicts to be managed Risk: Scheduling conflicts of employee rosters has an effect on < ..... > resulting in < ..... >	40%	7	2.8	2

Task 6: Risk Management

	Low Impact	High Impact
High Probability	High / Low	High / High
Low Probability	Low / Low	Low / High

Avoid

Transfer

Mitigate

Accept

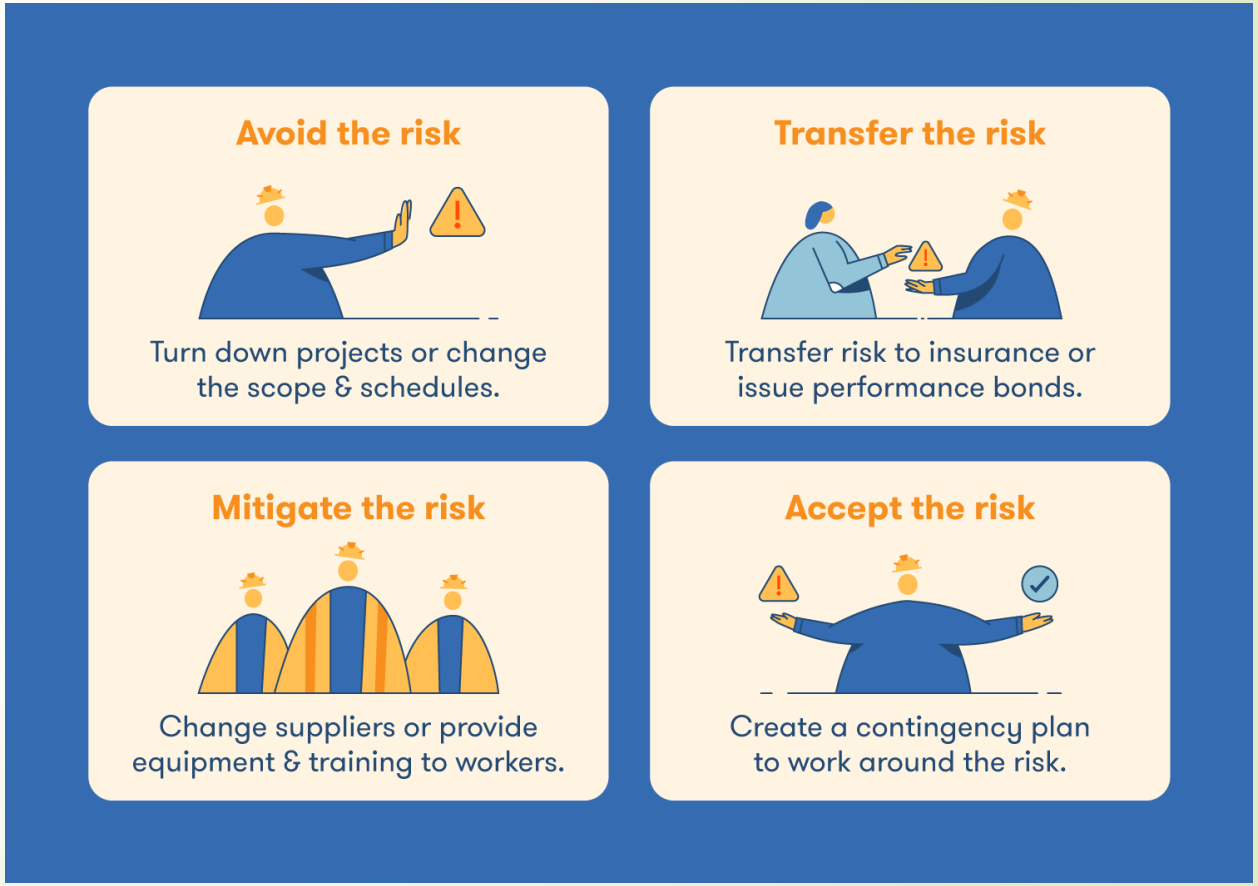
Classifying or categorizing risks (qualitative)

1. Low probability, low impact risks
   - Unlikely to happen, if they did they would cause little harm
   - Not worth spending project resources on
2. High probability, low impact
   - Can grow into large troublesome risks
   - Monitor and intercept before they get out of hand
3. High probability, high impact
   - Must be neutralised instantly
   - Spend resources addressing these carefully
4. Low probability, high impact
   - Not a problem if you know where they are
   - Try to avoid with care!

Risk response

Risk on project	Risk response strategy	Examples
Probability: high Impact: high	Avoid. Transfer. Mitigate	• Reduce scope • Add resources • Risk averse
Probability: low Impact: high	Avoid, transfer	• Use of insurance • Performance bonds • Warranties • Guarantees
Probability: high Impact: low	Mitigate	• Decrease probability of occurrence • Decrease impact if risk occurs
Probability: low Impact: low	Accept	• Active acceptance • Passive acceptance • Contingency allowance

- How should we treat the risk of a falling tree?



Tutorial Quiz!

- You have 5 mins to complete the quiz