REQUIREMENTS IMPLEMENTATION

PART ONE: REQUIREMENT PROCESS IMPROVEMENT (CH31)

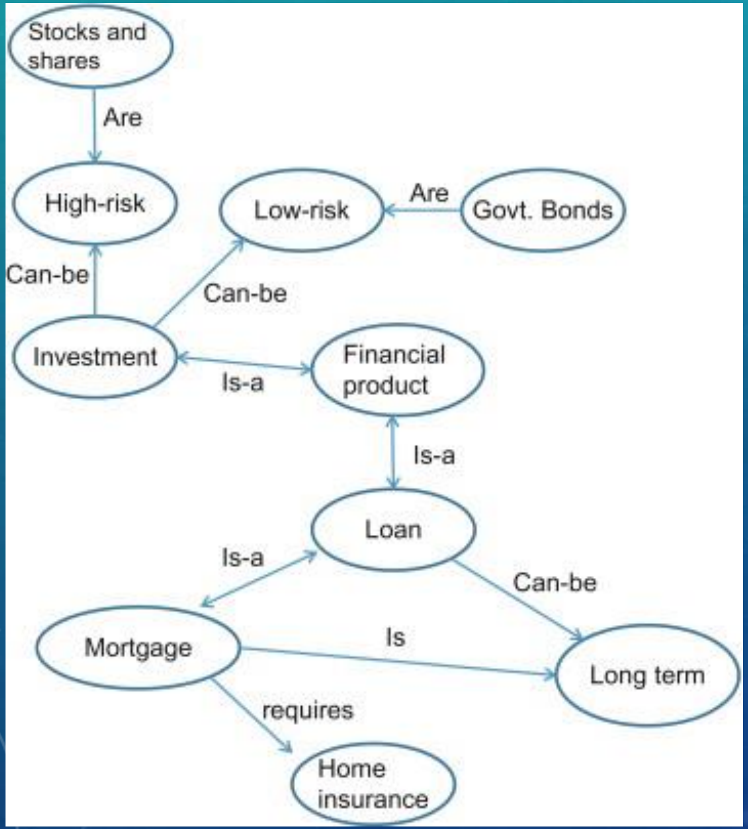
WHY IMPROVE?

- Improvement is about problem-solving. if the current requirement process seems to work well, there is no need to do so. Do not waste your time.
- If there were problems, then, ”learn from lessons” does make sense. It will allow you to be able to
  - Recall how the problems were encountered on current/previous projects (overtime, over budget, a lots last-minute changes, a lots of rework, etc.)
  - Figure out how the problems were corrected (good practice)
  - Anticipate and hence prevent problems that you might encounter on future projects (find causes first).
  - Adopt good practices that are more efficient and effective than those that are used in the previous projects, or try different practices.
- Improvement in requirement process will benefit all other parts of SDLC and stakeholders, because requirement process is related to other processes of software development projects and to all stakeholders.
- It could also have side effects on other project processes/stakeholders.
- It requires support/commitment from all parties, especially from ….

WHY: FROM PERSPECTIVE OF RELATIONSHIPS WITH OTHER PARTS OF A PROJECT

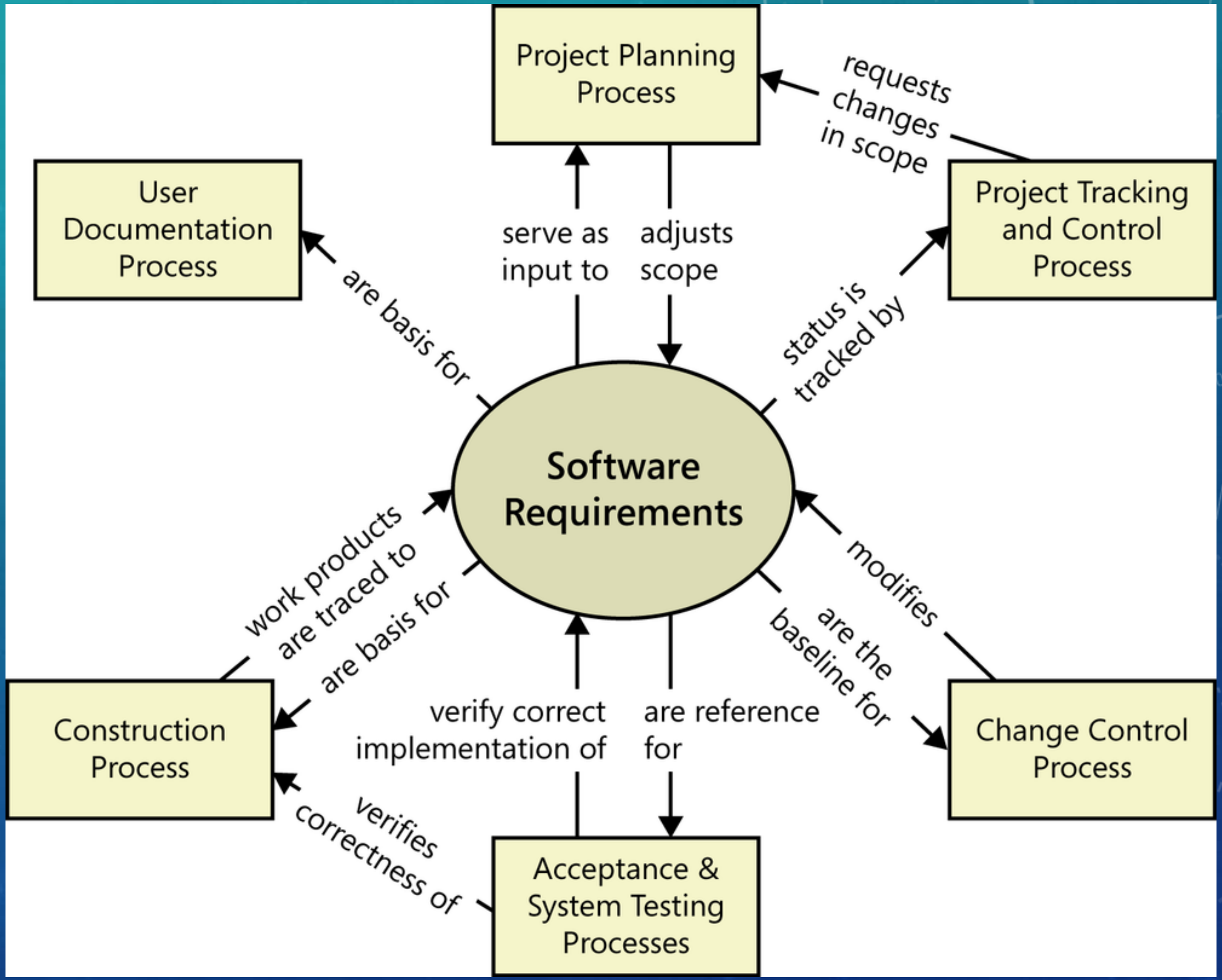
Background

- Semantic network represents knowledge and belongs to AI. It shows knowledge about objects and the relationship between objects known as semantic relations, in the format of network.
- Example: https://www.sciencedirect.com/topics/computerscience/semantic-network



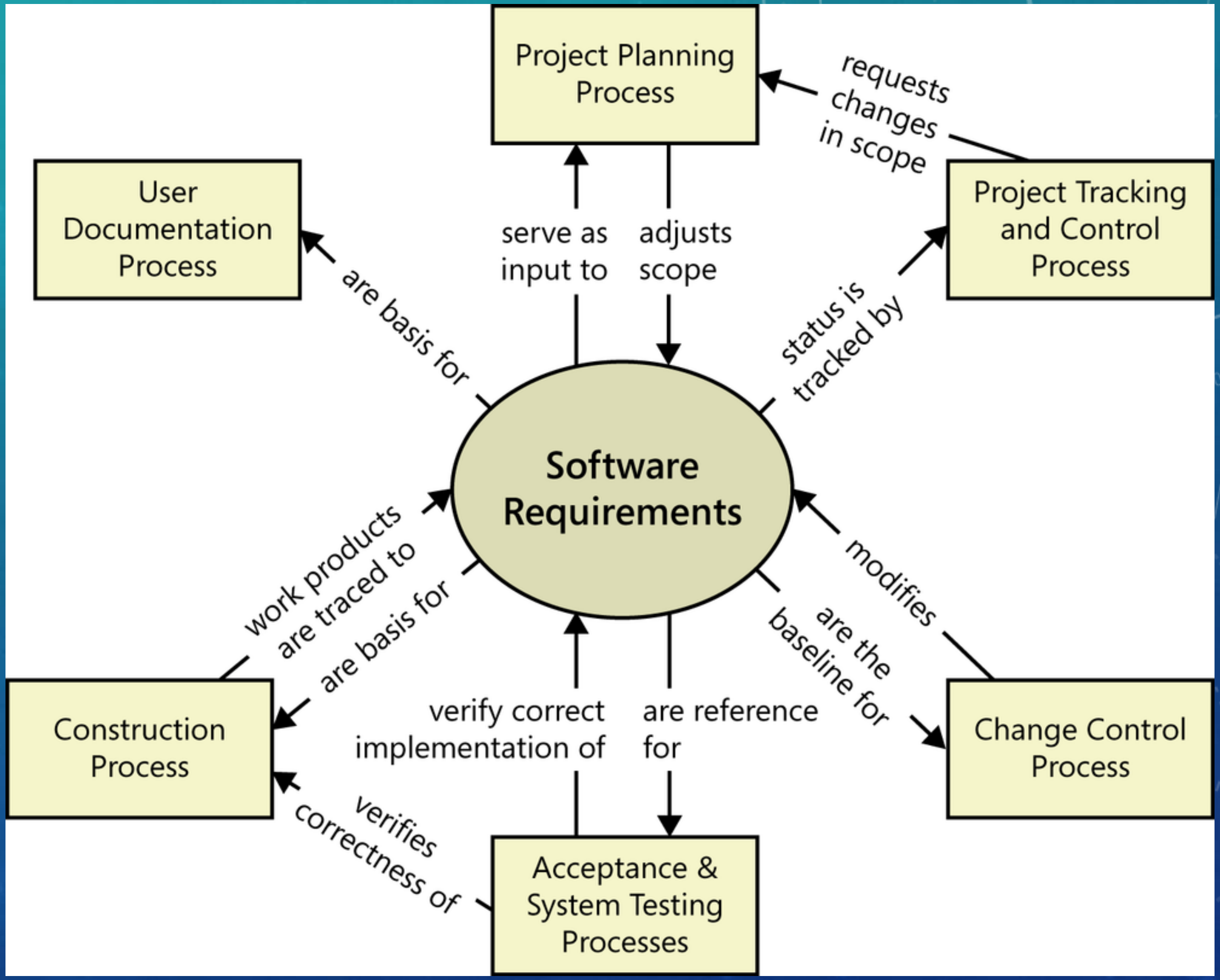
WHY: FROM PERSPECTIVE OF RELATIONSHIPS WITH OTHER PARTS OF A PROJECT

- Project planning: Requirements serve as the foundation of the project planning process. Based on the requirements, the planners select an appropriate software development life cycle and create resource and schedule estimates. Project planning also indicate some features may not be possible to deliver within the available bounds of resources and time. The planning process can lead to reductions in the project scope.
- Project tracking and control: Project tracking includes monitoring the project's status against requirements' status (refer to requirement status tracking).
- Change control: All subsequent changes and additions should be made through a defined change control process. Accepted changes will be added to the list of requirements.



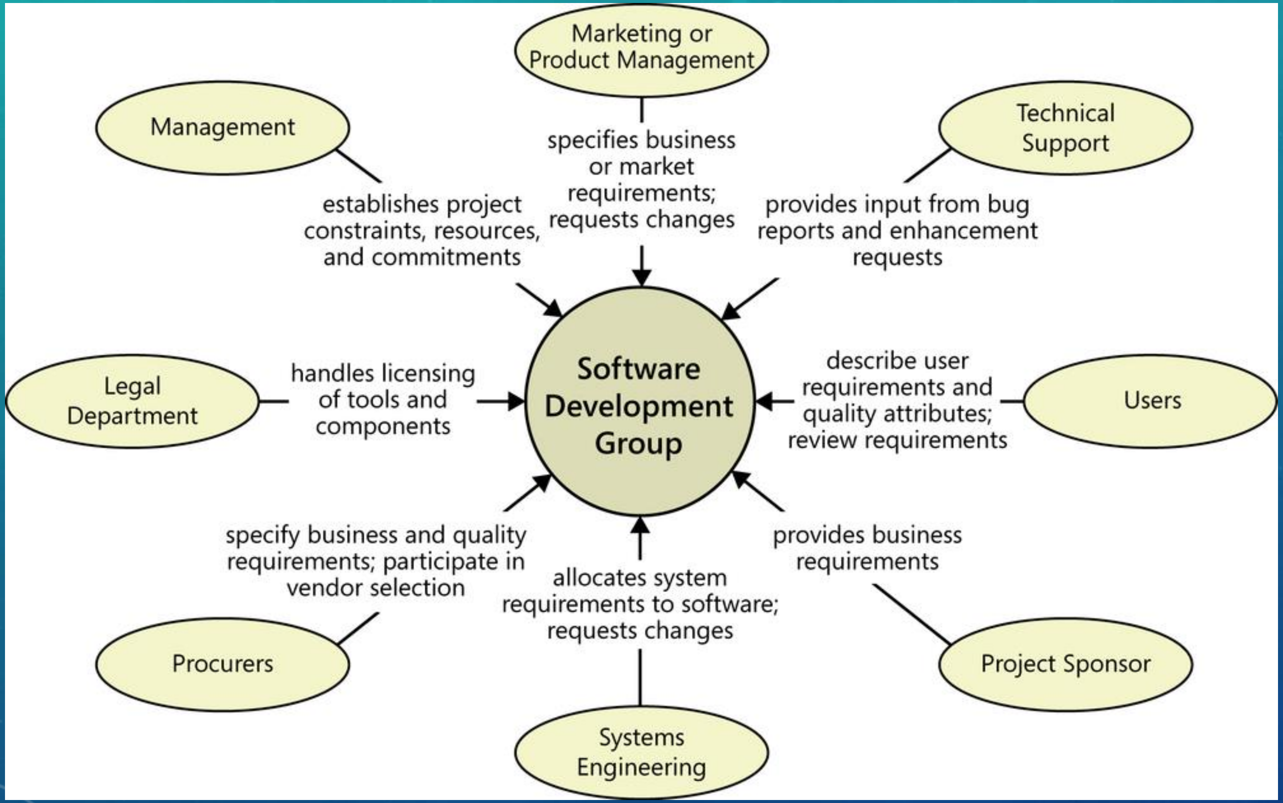
WHY: FROM PERSPECTIVE OF RELATIONSHIPS WITH OTHER PARTS OF A PROJECT

- Acceptance and system testing: User requirements and functional requirements are essential inputs to acceptance testing and system testing, respectively. The later informs requirement process on whether a requirement is verified or not.
- Construction: Requirements are the basis for the design and implementation work. Construction/development (or say design and coding) implements requirements.
- User documentation: The product's requirements provide input to the user documentation such as user manual.



HOW TO IMPROVE? -- RELATIONSHIP WITH STAKEHOLDERS

Contributions or inputs from different stakeholders to requirements


Marketing or Product Management
specifies business or market requirements; requests changes
Management
establishes project constraints, resources, and commitments
Technical Support
provides input from bug reports and enhancement requests
Legal Department
handles licensing of tools and components
Software Development Group
describe user requirements and quality attributes; review requirements
Users
specify business and quality requirements; participate in vendor selection
Procurers
allocates system requirements to software; requests changes
Systems Engineering
provides business requirements
Project Sponsor

There is a need to secure commitment from stakeholders to the improvement, in particular, the management of all stakeholders' organisation.

Resistance:

- People who are already too busy to get their project work done.
- An improved new process might be viewed as a barrier to make changes harder to make (don't bother, too much troubles.).
- Developers/managers view writing and reviewing requirements as bureaucratic time-wasters.

HOW TO IMPROVE? -- RELATIONSHIP WITH STAKEHOLDERS

This scenario shows some real concerns

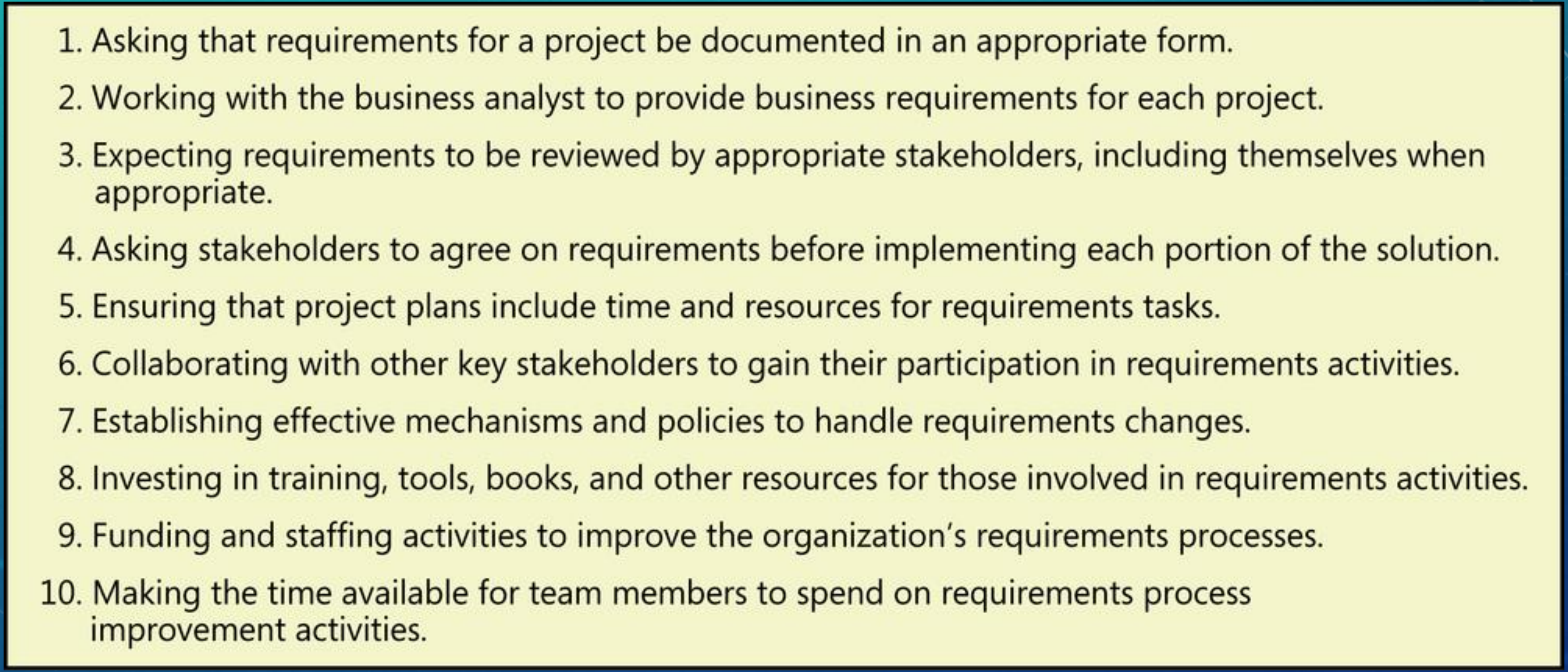
Everyone agreed that the last few projects had not gone smoothly. As the lead business analyst, Joanne knew that requirements issues had caused at least some of the problems. The BAs on the various projects varied greatly in their education and experience levels. They each used different approaches for developing and managing requirements, just doing the best they could based on what they knew. They each organized their requirements in different ways. Some teams followed effective requirements change processes, which reduced the turmoil in their projects, whereas others reacted to every change request that came along in a knee-jerk fashion. The frustration level was high all around.

Joanne had tried mentoring her less experienced BAs; some were more receptive to her input than others. Some of the teams in Joanne’s organization did do a good job on their requirements, and those projects suffered fewer headaches than those of the other teams. Joanne realized that it would be great to bring all of the teams up to a higher level of requirements performance. Maybe now the time was right to get serious about improving their requirements practices. But would the other BAs and their fellow team members play along? Was management truly committed to reducing the pain points? Would anything really change this time, or would this improvement initiative founder on the rocks of indifference, as the earlier ones had?

HOW TO IMPROVE? -- RELATIONSHIP WITH STAKEHOLDERS

Commitment from management is the key

- Signs of having their support:



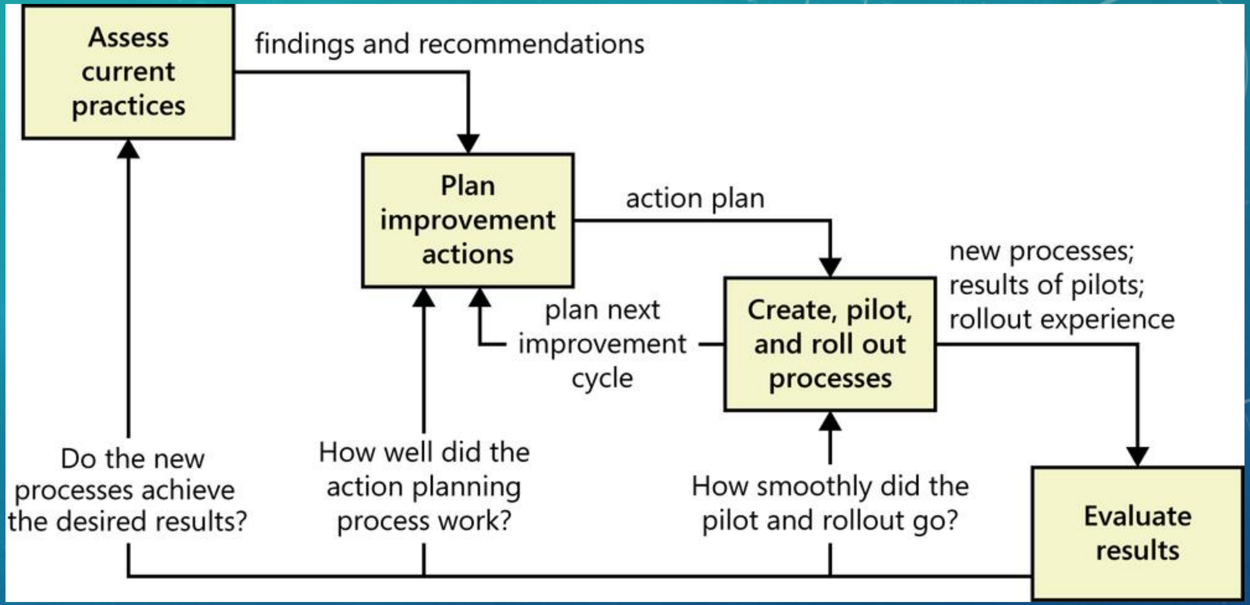
1. Asking that requirements for a project be documented in an appropriate form.
2. Working with the business analyst to provide business requirements for each project.
3. Expecting requirements to be reviewed by appropriate stakeholders, including themselves when appropriate.
4. Asking stakeholders to agree on requirements before implementing each portion of the solution.
5. Ensuring that project plans include time and resources for requirements tasks.
6. Collaborating with other key stakeholders to gain their participation in requirements activities.
7. Establishing effective mechanisms and policies to handle requirements changes.
8. Investing in training, tools, books, and other resources for those involved in requirements activities.
9. Funding and staffing activities to improve the organization’s requirements processes.
10. Making the time available for team members to spend on requirements process improvement activities.

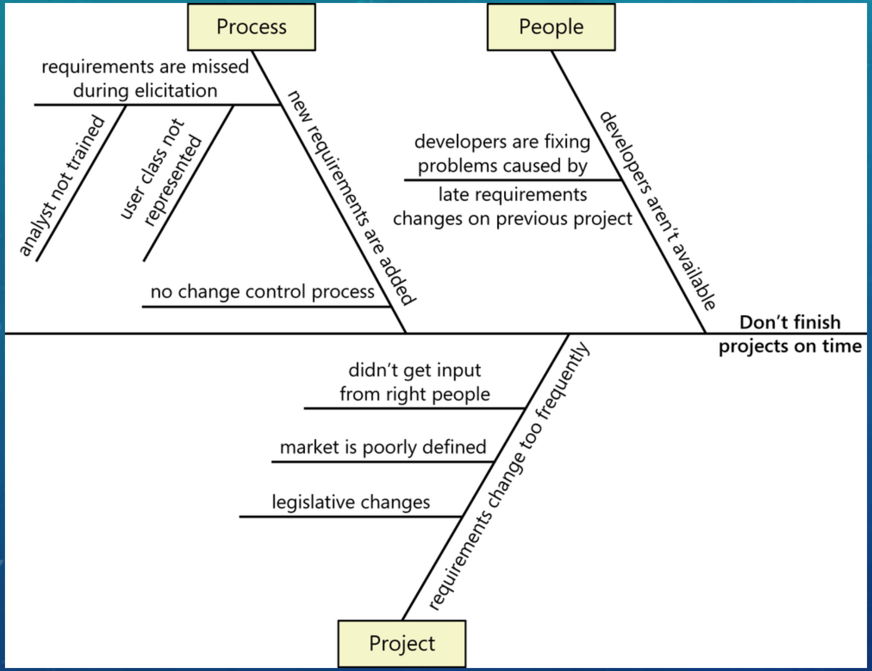
HOW TO IMPROVE? -- IMPROVEMENT PRINCIPLES

- Process improvement should be evolutionary and continuous
- People and organisations make improvements only when they have an incentive to do so
  - Project missed deadlines because requirements were more extensive than expected.
  - Developers worked a lot of overtime because of misunderstood or ambiguous requirements.
  - System test effort was wasted because testers didn’t understand what the product was supposed to do.
  - The right functionality was present, but users were dissatisfied because of sluggish performance, poor usability, or other quality shortcomings.
  - The company experienced high maintenance costs because customers requested many enhancements that could have been identified during requirements elicitation.
  - Requirement changes weren’t implemented appropriately during the course of the project, so the delivered solution did not meet the customer needs.
  - Edits to requirements were lost or overwritten because multiple BAs were working on them concurrently without a version control process.
  - Customers were not available to clarify and flesh out requirements.
  - Requirements-related issues were not resolved in a timely fashion, causing rework.
- Process improvements should be goal-oriented
- Treat your improvement activities as mini-projects

HOW TO IMPROVE? -- IMPROVEMENT PROCESS CYCLE

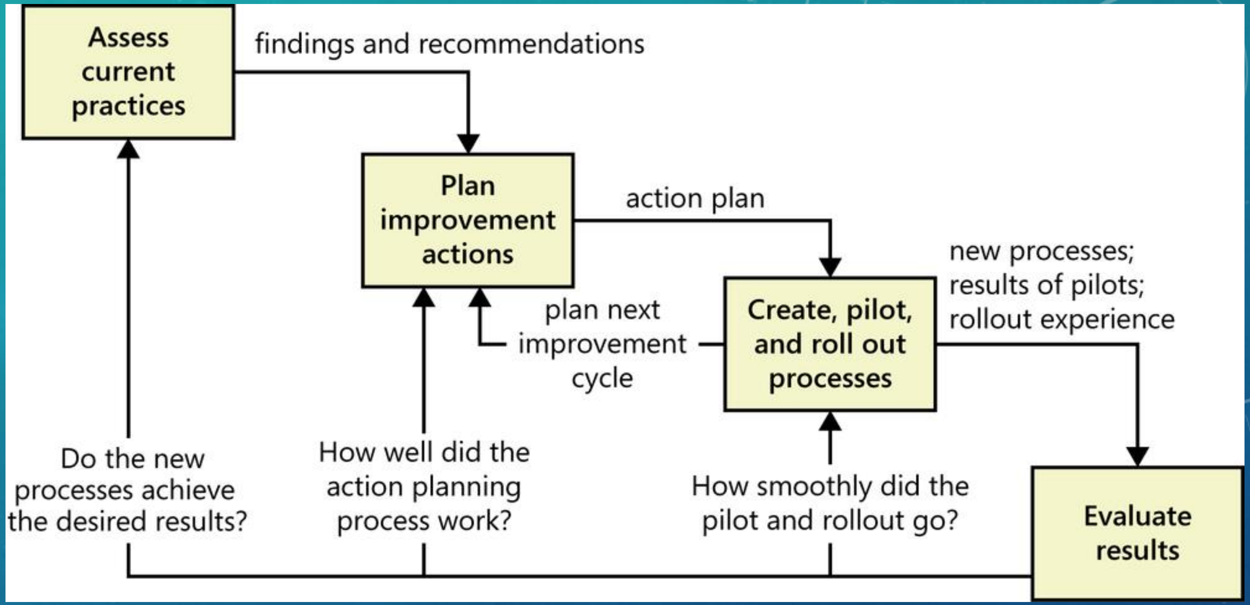
- Step 1: assess the current practices
  - Root-cause analysis (fishbone analysis) -- Root cause analysis involves asking "why" the problem exists several times in succession, each time probing for the reason that underlies the answer to the previous "why" question. (problem and cause are two different things.)


Assess current practices
findings and recommendations
Plan improvement actions
action plan
Create, pilot, and roll out processes
new processes; results of pilots; rollout experience
Evaluate results
plan next improvement cycle
Do the new processes achieve the desired results?
How well did the action planning process work?
How smoothly did the pilot and rollout go?


Process
People
Project
requirements are missed during elicitation
analyst not trained
user class not represented
new requirements are added
no change control process
developers are fixing problems caused by late requirements changes on previous project
developers aren't available
Don't finish projects on time
didn't get input from right people
market is poorly defined
legislative changes
requirements change too frequently

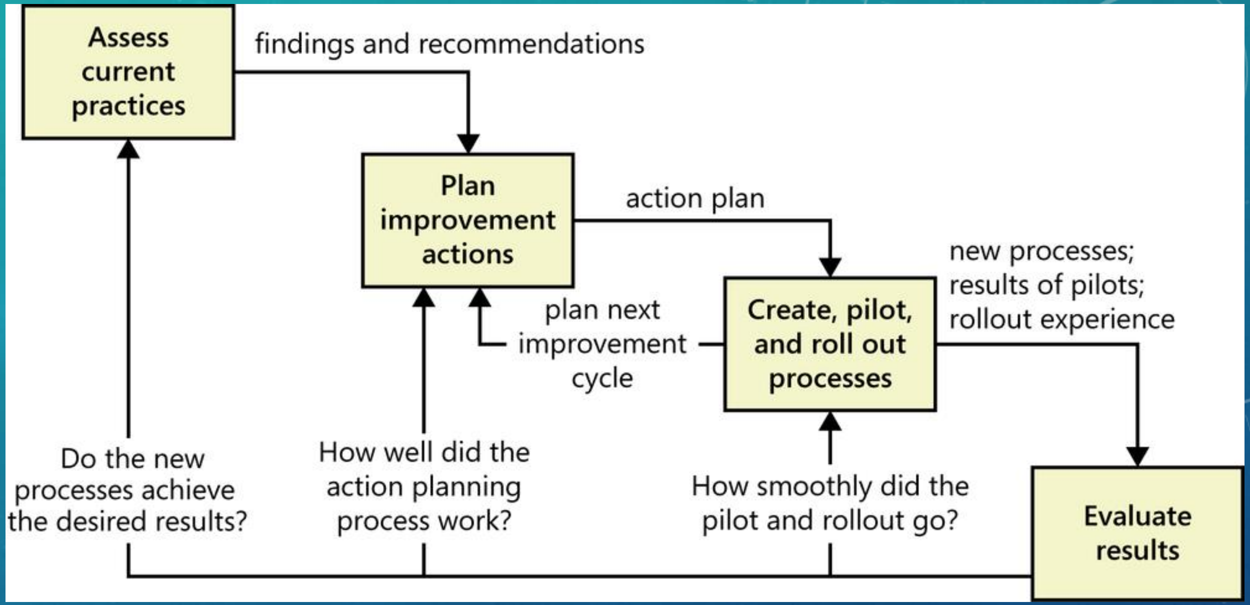
HOW TO IMPROVE? -- IMPROVEMENT PROCESS CYCLE

- Step 2: Planning
  - Draft an improvement proposal.
  - Review and revise the proposal.
  - Pilot the improvement actions in Project A.
  - Revise the change and the actions based on feedback from the pilot.
  - Evaluate problem-tracking tools, and select one to support the change control process.
  - Procure the problem-tracking tool, and customize it to support the change control process.


Assess current practices
findings and recommendations
Plan improvement actions
action plan
Create, pilot, and roll out processes
new processes; results of pilots; rollout experience
plan next improvement cycle
Evaluate results
Do the new processes achieve the desired results?
How well did the action planning process work?
How smoothly did the pilot and rollout go?

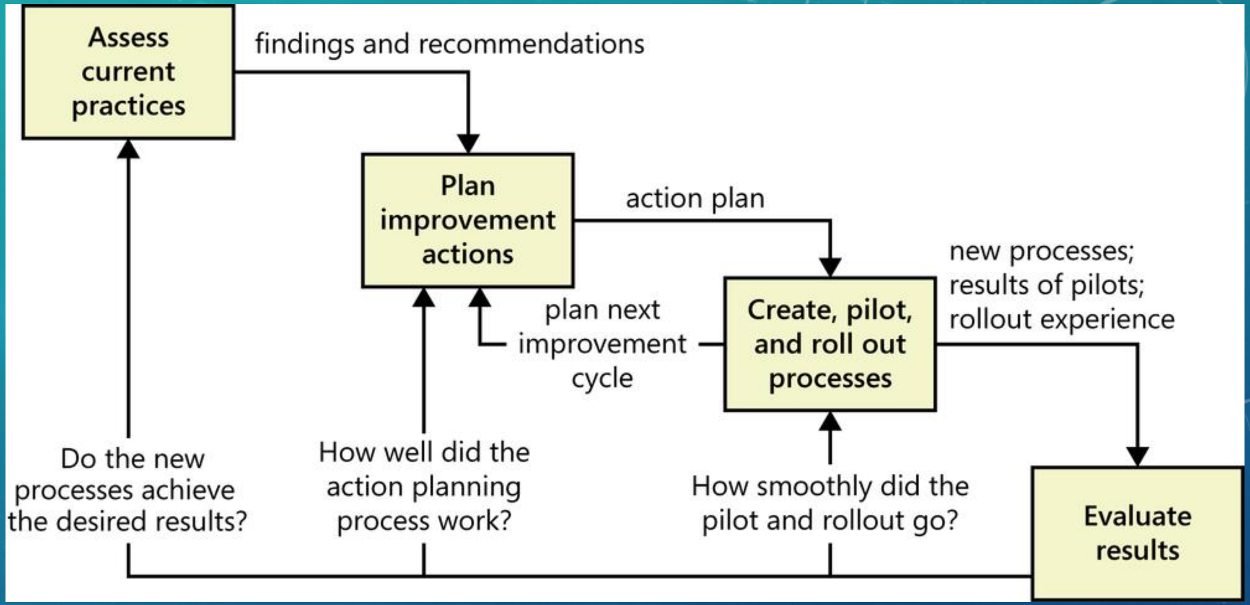
HOW TO IMPROVE? -- IMPROVEMENT PROCESS CYCLE

- Step 3: implementing Pilot is a good idea.
  - Select pilot participants who will give the new approaches a fair try and provide helpful feedback. These participants could be either allies or sceptics but shouldn't strongly oppose the improvement effort.
  - Quantify the criteria which the team will use to evaluate the pilot's results.
  - Identify the stakeholders who need to be informed about the pilot and why it is being performed.
  - Consider piloting portions of the new processes on different projects. This engages more people in trying new approaches, which increases awareness, feedback, and buy-in.
  - Roll out the new actions and tool to the company.


Assess current practices
findings and recommendations
Plan improvement actions
action plan
Create, pilot, and roll out processes
new processes; results of pilots; rollout experience
plan next improvement cycle
Evaluate results
Do the new processes achieve the desired results?
How well did the action planning process work?
How smoothly did the pilot and rollout go?

HOW TO IMPROVE? -- IMPROVEMENT PROCESS CYCLE

- Step 4: Evaluating
    - A critical step is to assess whether the new processes are yielding the desired results.
    - As part of the evaluation, ask pilot participants how they would feel if they had to go back to their former ways of working.


Assess current practices
findings and recommendations
Plan improvement actions
action plan
Create, pilot, and roll out processes
new processes; results of pilots; rollout experience
plan next improvement cycle
Evaluate results
Do the new processes achieve the desired results?
How well did the action planning process work?
How smoothly did the pilot and rollout go?

Cycle!!
This is not a linear process. It is an iterative process.
- Step 3 can lead to plan next improvement.
- Step 4 can cause improvements at all previous steps.

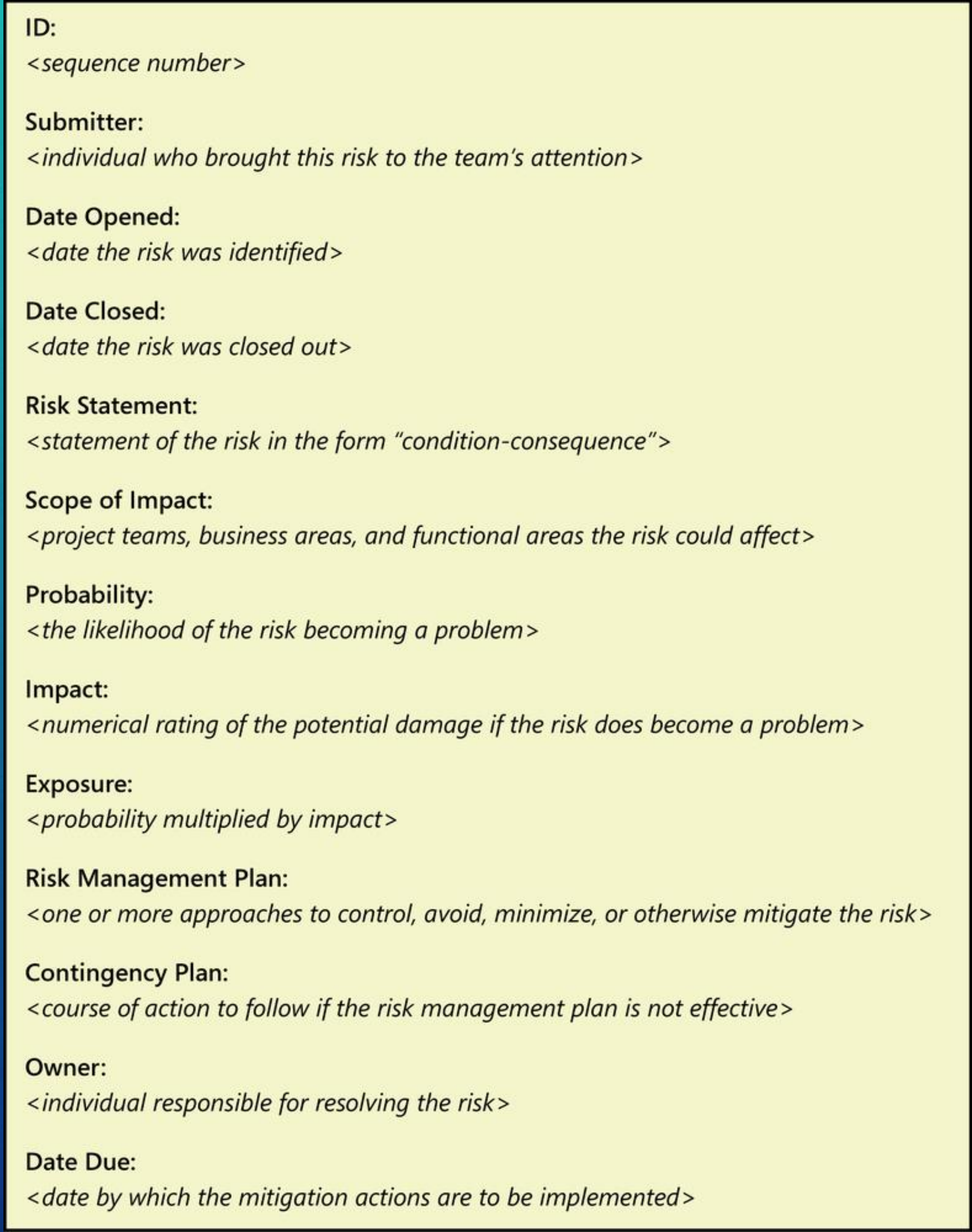
PART TWO: MANAGING RISKS (CH32)

WHAT IS A RISK?

- The possibility of something bad happening (Cambridge dictionary)

- A risk is a condition that could cause some loss or otherwise threaten the success of a project.
- Risk and reward -- The risk-reward relationship is based on the concept that the higher the risk of loss of principal for an investment, the greater the potential reward of an increase in the principal or higher yield on the principal.
- In software engineering, risks can lead project failure.
- Award is simply the delivery of software products on time and within the budget, leading to a great profit.

- Node down risks in a logbook



ID:
<sequence number>

Submitter:
<individual who brought this risk to the team's attention>

Date Opened:
<date the risk was identified>

Date Closed:
<date the risk was closed out>

Risk Statement:
<statement of the risk in the form "condition-consequence">

Scope of Impact:
<project teams, business areas, and functional areas the risk could affect>

Probability:
<the likelihood of the risk becoming a problem>

Impact:
<numerical rating of the potential damage if the risk does become a problem>

Exposure:
<probability multiplied by impact>

Risk Management Plan:
<one or more approaches to control, avoid, minimize, or otherwise mitigate the risk>

Contingency Plan:
<course of action to follow if the risk management plan is not effective>

Owner:
<individual responsible for resolving the risk>

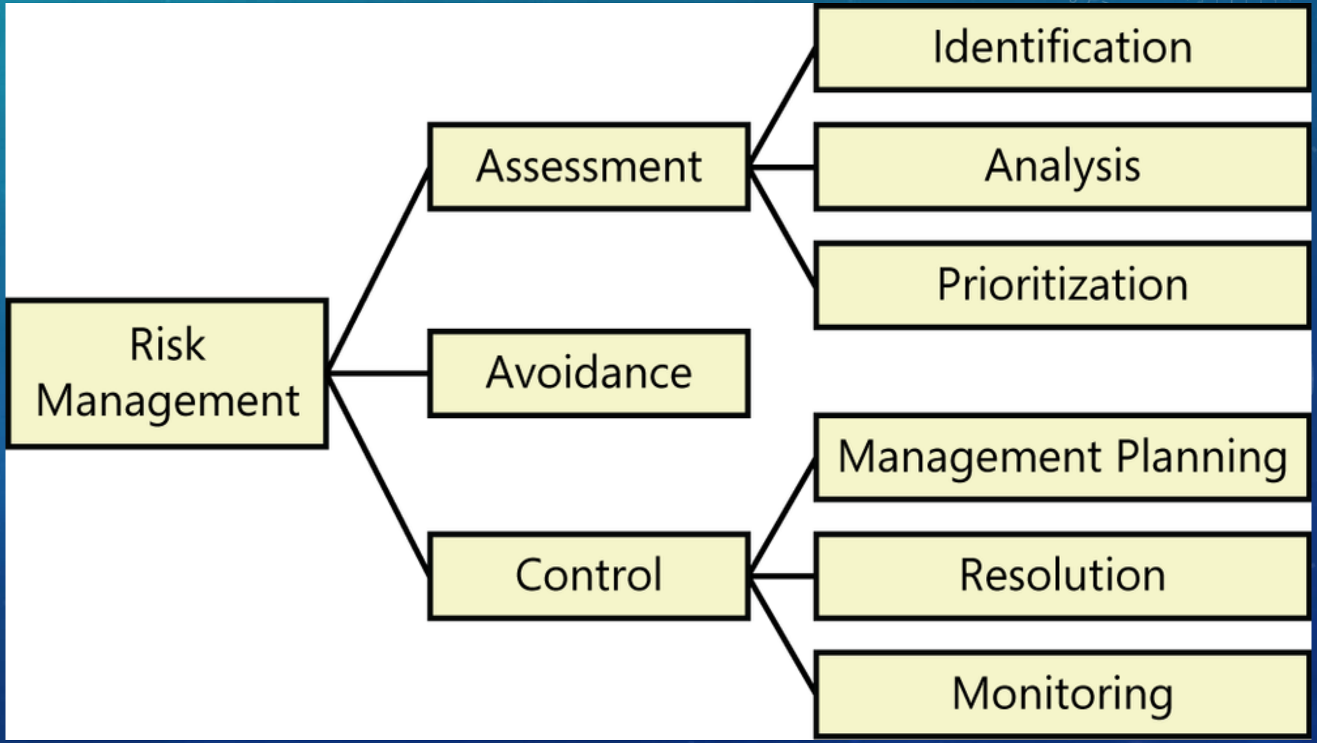
Date Due:
<date by which the mitigation actions are to be implemented>

RISK MANAGEMENT

Software risk management is a balance of risk and reward

Risk management process

- Step 1: Identify the Risk. uncover, recognize and describe risks that might affect your project or its outcomes.
- Step 2: Analyse the risk. determine the likelihood and consequence of each risk.
- Step 3: Evaluate or Rank the Risk. evaluate or rank the risk by determining the risk magnitude, which is the combination of likelihood and consequence.
- Step 4: Handle the Risk. assess the highest ranked risks and set out a plan to treat or modify these risks to achieve acceptable risk levels. (contingent plan and risk migration)
- Step 5: Monitor and Review the risk. take your Project Risk Register and use it to monitor, track and review risks



RISKS IN REQUIREMENT ENGINEERING

In requirements elicitation

- Scope creep
  - Caused by – Stakeholders do not share their understandings on what the product is supposed to be and to do at the beginning of a project but gradually add them on.
  - How to deal with – Develop product vision and project scope document and use it as the reference to interpret requirements, strictly follow CCB
- Expectation gap among stakeholders
  - Caused by – Chaos in customer engagement
  - How to handle – Identify a representative for each customer group and description of each use case in plain language
- Incompleteness and incorrectness of requirements specifications
  - Caused by – Customers themselves are not clear about business objectives
  - How to handle – Mapping user requirements with business objectives and having customers to decide acceptance criteria

RISKS IN REQUIREMENT ENGINEERING

In requirements elicitation (continue)

- Misgauge market response to innovative products
  - Caused by – The products are the first of their kind
  - How to handle – Market research, prototype and user focus group to obtain early feedback
- Neglect non-functional requirements
  - Caused by – Pay too much attention to functional requirements
  - How to handle – Query customer about quality attributes and document the non-functional requirements
- Some customers are unhappy with requirement documents
  - Caused by – Lack of communications and agreement on requirements
  - How to handle – Focus on primary customer who are normally the users

RISKS IN REQUIREMENT ENGINEERING

In requirements elicitation (continue)

- Unstated requirements
  - Caused by – Customers often hold implicit expectations and suppose BA know the hidden expectations
  - How to handle – Use open-end questions to obtain more thoughts from customers
- Solutions presented as needs
  - Caused by – Customers have their thoughts about solutions
  - How to handle – UML use case diagram
- Distrust between the business and the development team
  - Caused by – Feeling the threats of possible redundancy
  - How to handle -- ?

RISKS IN REQUIREMENT ENGINEERING

In requirements analysis

- Requirements not prioritised properly
  - Caused by – entirely recording what customers say and hence lack of analysis
  - How to handle – appropriate trade-off decisions
- Technically difficult features
  - Caused by – insufficient feasibility studies
  - How to handle – tracking requirements implementation to identify those that failing behind schedule as early as possible
- Longer learning time on unfamiliar technologies, methods, languages, tools, or hardware
  - Caused by – unfamiliar technologies, methods, languages, tools, or hardware
  - How to handle – identify high-risk components early to allow development team to learn new stuff

RISKS IN REQUIREMENT ENGINEERING

In requirement specification

- Different requirements understandings
  - Caused by – Different interpretations of the requirements by developers and customers
  - How to handle – Peer review by customers, developers and testers
- Open issues (TBD)
  - Caused by – Time pressure
  - How to handle – Clarifying who has the responsibility for closing the issues
- Ambiguous terminology
  - Caused by – natural languages
  - How to handle – Glossary to define business and technological terms
- Design in requirements
  - Caused by – Customers thought they new solutions
  - How to handle – Asking customers to emphasise their needs

RISKS IN REQUIREMENT ENGINEERING

In requirement validation

- Unvalidated requirements
  - Caused by – Lengthy requirement documents
  - How to handle – Make sure V-model is followed and formal/informal peer review.
- Lack of experience in inspection
  - Caused by – It is inevitable that new people will join inspection
  - How to handle – Training

In requirement management

- Changing requirements – CCB
- Requirement change control – CCB
- Unimplemented requirements – requirement tracing
- Expending project scope – CCB