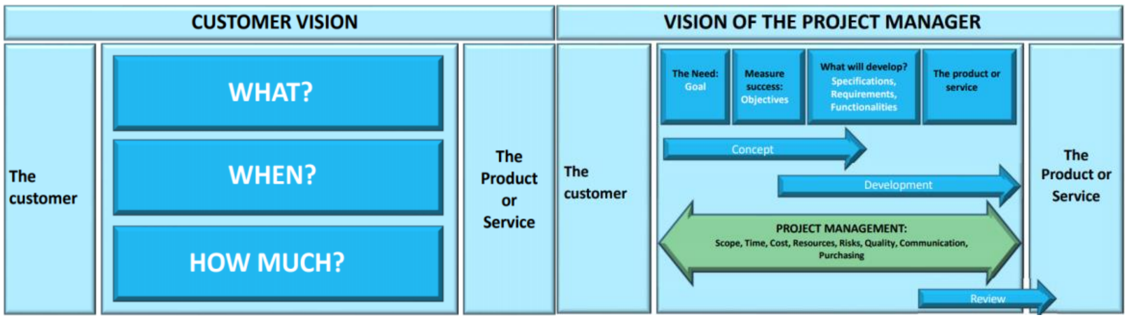
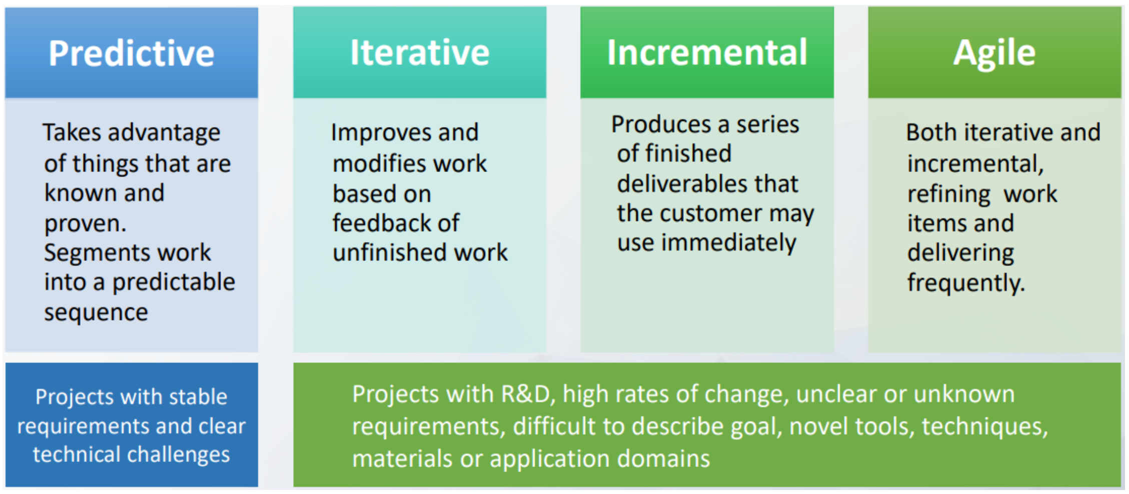
Introduction

How are projects managed?

Projects are often too complex to be managed without control and structure. A structured methodology simplifies entries and project management becomes achievable.


CUSTOMER VISION
VISION OF THE PROJECT MANAGER
The customer
WHAT?
WHEN?
HOW MUCH?
The Product or Service
The customer
The Need: Goal
Measure success: Objectives
What will develop? Specifications, Requirements, Functionalities
The product or service
Concept
Development
PROJECT MANAGEMENT: Scope, Time, Cost, Resources, Risks, Quality, Communication, Purchasing
Review
The Product or Service



Predictive	Iterative	Incremental	Agile
Takes advantage of things that are known and proven. Segments work into a predictable sequence	Improves and modifies work based on feedback of unfinished work	Produces a series of finished deliverables that the customer may use immediately	Both iterative and incremental, refining work items and delivering frequently.
Projects with stable requirements and clear technical challenges	Projects with R&D, high rates of change, unclear or unknown requirements, difficult to describe goal, novel tools, techniques, materials or application domains		

Projects come in many shapes and there are a variety of ways to undertake them. Project teams need awareness of the characteristics and options available to select the approach most likely to be successful for the situation.

This practice guide refers to four types of life cycles, defined as follows:

◆ Predictive life cycle. A more traditional approach, with the bulk of planning occurring upfront, then executing in a single pass; a sequential process.

◆ Iterative life cycle. An approach that allows feedback for unfinished work to improve and modify that work.

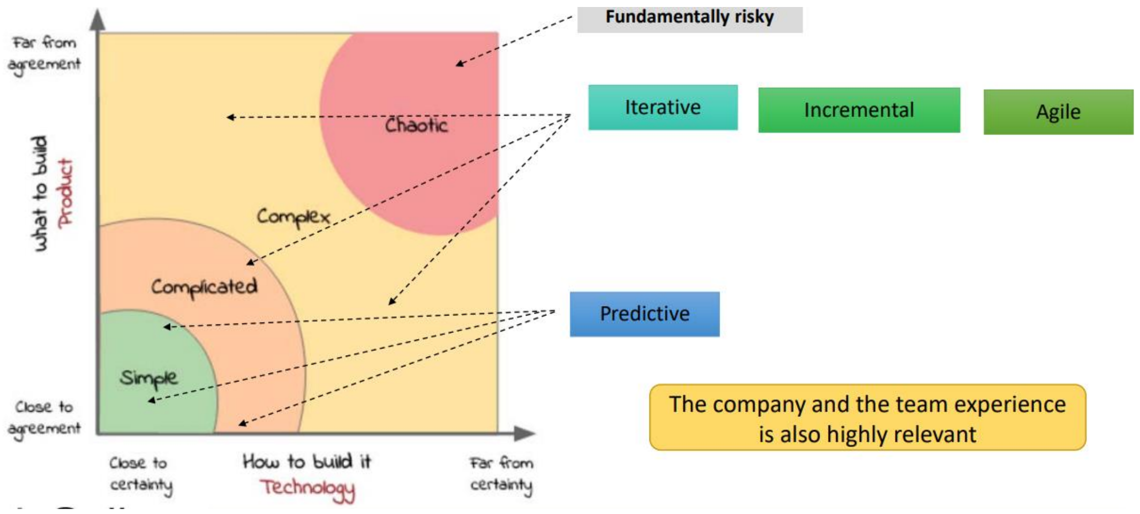
◆ Incremental life cycle. An approach that provides finished deliverables that the customer may be able to use immediately.

◆ Agile life cycle. An approach that is both iterative and incremental to refine work items and deliver frequently.

It is important to note that all projects have these characteristics—no project is completely devoid of considerations around requirements, delivery, change, and goals. A project's inherent characteristics determine which life cycle is the best fit for that project.

Another way to understand how project life cycles vary is by using a continuum ranging from predictive cycles on one end, to agile cycles on the other end, with more iterative or incremental cycles in the middle.

Work varies from work with clear procedures that proved successful on similar projects in the past to work with high rates of change, complexity, and risk.



Teams can plan and manage projects with clear and stable requirements and clear technical challenges with little difficulty. However, as the uncertainty in the project increases, the likelihood of changes, wasted work, and rework also increases, which is costly and time consuming.

These iterative, incremental, and agile approaches work well for projects that involve new or novel tools, techniques, materials, or application domains. (Refer to Section 3 on Life Cycle Selection). They also work well for projects that:

- Require research and development. Have high rates of change.
- Have unclear or unknown requirements, uncertainty, or risk; or
- Have a final goal that is hard to describe.

By building a small increment and then testing and reviewing it, the team can explore uncertainty at a low cost in a short time, reduce risk, and maximize business value delivery. This uncertainty may be centred on suitability and requirements (is the right product being built?); technical feasibility and performance (can this product be built this way?); or process and people (is this an effective way for the team to work?). All three of these characteristics—product specification, production capability, and process suitability—typically have elements of high uncertainty.

However, iterative, and incremental approaches have their limits of applicability. When both technology uncertainty and requirements uncertainty are very high (the top right of Figure 2-5), the project moves beyond complex to chaotic. For the project to become reliably possible, it needs one of the variables (uncertainty or disagreement) to be contained.

A project's inherent characteristics determine which life cycle is the best fit for that project.

Predictive lifecycle

Predictive life cycles expect to take advantage of high certainty around firm requirements, a stable team, and low risk. As a result, project activities often execute in a serial manner.

To achieve this approach, the team requires detailed plans to know what to deliver and how. These projects succeed when other potential changes are restricted (e.g., requirements change; project team members change what the team delivers). Team leaders aim to minimize change for the predictive project.

When the team creates detailed requirements and plans at the beginning of the project, they can articulate the constraints. The team can then use those constraints to manage risk and cost. As the team progresses through the detailed plan, they monitor and control changes that might affect the scope, schedule, or budget.

By emphasizing a departmentally efficient, serialized sequence of work, predictive projects do not typically deliver business value until the end of the project. If the predictive project encounters change or disagreements with the requirements, or if the technological solution is no longer straightforward, the predictive project will incur unanticipated costs.

Iterative lifecycle

Iterative life cycles improve the product or result through successive prototypes or proofs of concept. Each new prototype yields new stakeholder feedback and team insights. Then, the team incorporates the new information by repeating one or more project activities in the next cycle. Teams may use timeboxing on a given iteration for a few weeks, gather insights, and then rework the activity based on those insights. In that way, iterations help identify and reduce uncertainty in the project.

Projects benefit from iterative life cycles when complexity is high, when the project incurs frequent changes, or when the scope is subject to differing stakeholders' views of the desired final product. Iterative life cycles may take longer because they are optimized for learning rather than speed of delivery.

Incremental lifecycle

Some projects optimize for speed of delivery. Many businesses and initiatives cannot afford to wait for everything to be completed; in these cases, customers are willing to receive a subset of the overall solution. This frequent delivery of smaller deliverables is called an incremental life cycle.

Incremental life cycles optimize work for delivering value to sponsors or customers more often than a single, final product. Teams plan initial deliverables before beginning their work, and they begin working on that first delivery as soon as possible. As the project continues, the team may deviate from the original vision. The team can manage the deviations because the team delivers value sooner. The degree of change and variation is less important than ensuring customers get value sooner than at the end of the project.

Providing a customer, a single feature or a finished piece of work is an example of the incremental approach.

For example, builders may want to show a finished room or floor of a building before they continue with the remainder of the building. In that case, they may complete a floor with fixtures, paint, and everything else intended for the finished floor before proceeding to the next floor. The customer can see and approve of the style, colour, and other details, allowing adjustments to be made before further investments of time and money are made. This reduces potential rework and/or customer dissatisfaction.

viable product (MVP) to a subset of customers. The customers' feedback helps the team to learn what they need to provide for subsequent delivery of the final finished feature.

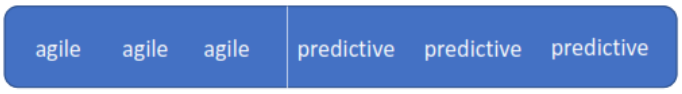
Agile teams, as a key differentiator, deliver business value often. As the product adds a broader set of features and a broader range of consumers, we say it is delivered incrementally.

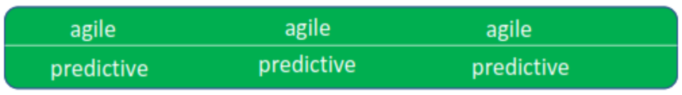
Agile lifecycle

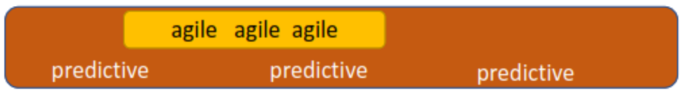
In an agile environment, the team expects requirements to change. The iterative and incremental approaches provide feedback to better plan the next part of the project. However, in agile projects, incremental delivery uncovers hidden or misunderstood requirements. Figure 3-5 illustrates two possible ways to achieve incremental delivery so the project aligns with customer needs and can be adapted, as necessary.

In iteration-based agile, the team works in iterations (timeboxes of equal duration) to deliver completed features. The team works on the most important feature, collaborating as a team to finish it. Then the team works on the next most important feature and finishes it. The team may decide to work on a few features at a time, but the team does not address all of the work for the iteration at once (i.e., does not address all of the requirements, followed by all of the analyses, etc.). Agile life cycles are those that fulfil the principles of the Agile Manifesto. Customer satisfaction increases with early and continuous delivery of valuable products. Moreover, an incremental deliverable that is functional and provides value is the primary measure of progress. Agile life cycles combine both iterative and incremental approaches to adapt to high degrees of change and deliver project value more often.

Agile lifecycle







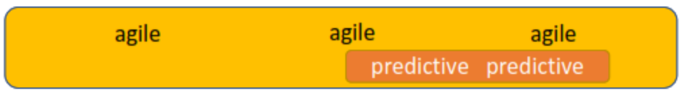


Figure 3-6 depicts the basic, pure approaches to project types that combine to form a hybrid model. The early processes utilize an agile development life cycle, which is then followed by a predictive rollout phase. This approach can be used when there is uncertainty, complexity, and risk in the development portion of the project that would benefit from an agile approach, followed by a defined, repeatable rollout phase that is appropriate to undertake in a predictive way, perhaps by a different team. An example of this approach is the development of a new high-tech product followed by rollout and training to thousands of users.

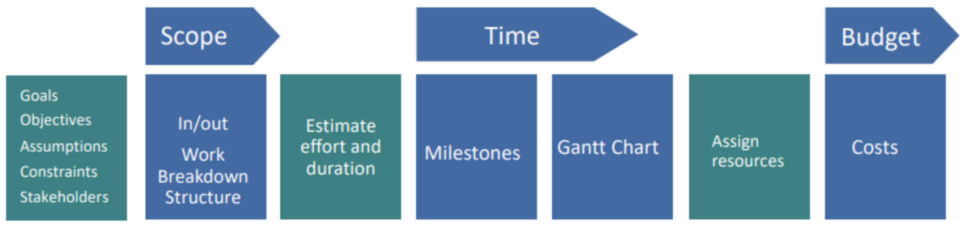
In Figure 3-7, a combination of both predictive and agile approaches is used in the same project. Perhaps the team is incrementally transitioning to agile and using some approaches like short iterations, daily stand-ups, and retrospectives, but other aspects of the project such as upfront estimation, work assignment, and progress tracking are still following predictive approaches.

Using both predictive and agile approaches is a common scenario. It would be misleading to call the approach agile since it clearly does not fully embody the agile mindset, values, and principles. However, it would also be inaccurate to call it predictive since it is a hybrid approach.

Figure 3-8 shows a small agile element within a chiefly predictive project. In this case, a portion of the project with uncertainty, complexity, or opportunity for scope creep is being tackled in an agile way, but the remainder of the Project is being managed using predictive approaches. An example of this approach would be an engineering firm that is building a facility with a new component. While most of the project may be routine and predictable, like many other facilities projects the organization has undertaken before, this project incorporates a new roofing material. The contractor may plan for some small-scale installation trials on the ground first to determine the best installation method and to uncover issues early while there is plenty of time to solve them and incrementally improve processes through experimentation and adaptation.

Figure 3-9 depicts a largely agile approach with a predictive component. This approach might be used when a particular element is non-negotiable or not executable using an agile approach. Examples include integrating an external component developed by a different vendor that cannot or will not partner in a collaborative or incremental way. A single integration is required after the component is delivered.

Scope



Definition of Scope: If it is not included in the Scope, it does not exist

• Identifies the necessary work to deliver the project result / product

• Translates objectives into deliverables

• It is the basis from what to establish the programme, the budget, and the resources requirements

• It must include management activities

Project Requirements

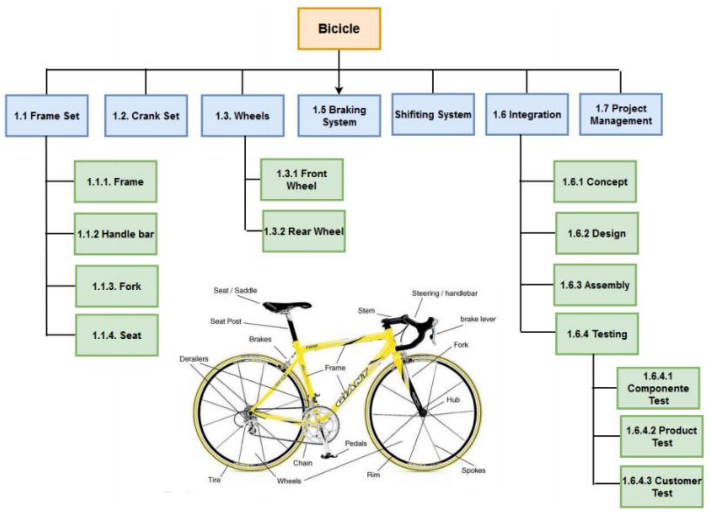
• Characteristics, needs, wishes and expectations that stakeholders want to include within the project and/or product

• Physical characteristics

• Functionality for users / clients

• Quality, security, etc.

• Deliver, budget



INCLUDED: Work that must be done to develop and deliver the product

EXCLUDED: Work that is not included in the project even though it also helps in defining the product we want to develop

A **Work Breakdown Structure** is a deliverable oriented hierarchical decomposition of the work to be executed by the project team to accomplish the project objectives and create the required deliverables. It defines and organizes the total scope of a project, using a hierarchical tree structure.

■ Includes everything that must be done in a Project

■ A WBS is NOT a project plan, a schedule, nor a chronological listing Only answers the "WHAT" must be done, not how or when Work Breakdown Structure (WBS)

Any work excluded from the WBS is considered outside the Project Scope "The 100% rule"

Ask the questions:

- What work must be done to create the product/service and its parts?
- The integration of all elements in the last level allow to generate the complete product or service?

Last level must be deliverables:

- NOT "Document the system", but instead "System manual"
- NOT "Organize the team", but "Team organization chart"

Work Breakdown Structure – Checklist

- The top element of the WBS is the overall deliverable of the project, and all stakeholders agree with it
- The First two levels of the WBS (the root node and level 2) define a set of planned outcomes that collectively and exclusively represent 100% of the project scope
- The WBS elements are defined in terms of outcomes or results
- Each WBS element has an identification number assigned (code) which identifies its relative position within the structure
- There is no overlap in scope definition between two elements of a WBS
- The WBS is not a chronological listing, and it is not a project plan or a project schedule

Schedule

Time Management

• Identify the specific tasks required to complete the project deliverables

Defining the activities

• Identify and document the relationships between project activities

Sequencing the activities

• Estimate the type and quantity of materials, equipment, people, and supplies

Estimate the resources for carrying out the activities

• Establish the quantity of work periods required to complete each activity

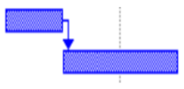
Estimate duration of activities

• Analyse the sequence of the activities, its duration, the resources requirements, and the schedule restrictions to create the project schedule

Develop the project schedule

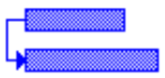
• Control project progress and manage changes

Simply translate the components from the Work Breakdown Structure into tasks



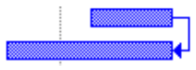
Finish-to-Start (FS): The work of task 2 can start only after all the work of task 1 is finished.

Examples: “Supper starts when cooking is complete”, “Concrete pouring after forms are built”



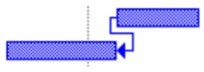
Start-to-Start (SS): The work of task 2 cannot start until task 1 starts.

Examples: “Mixing of chemicals cannot start until air quality testing starts”



Finish-to-Finish (FF): The work of task 2 cannot start until task 1 starts.

Examples: “Fracking cannot finish until pressure testing is complete”



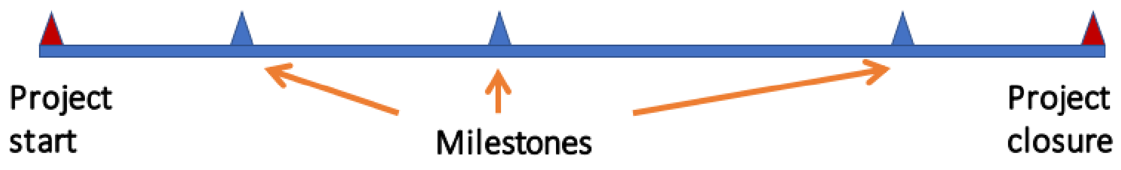
Start-to-Finish (SF): The start date of task 1 determines the finish date of task 2.

Examples: “Concrete Pouring cannot finish until Heating has started”

Milestones

To ensure that the project progresses satisfactorily, management checkpoints or milestones should be clearly defined with planned dates to measure progress.

Key Stakeholders use them to approve the completion of a phase or milestone and as go / no-go decision points to proceed with the project. Ensure that the products and services delivered meet the project objectives.



Popular Types of Milestones

- Completion of a deliverable: By far the most popular choice for a milestone because it represents tangible evidence of progress towards the project's goals.
- Rates of Completion: commonly used milestones when the project involves repetition without sequential advancement to the next stage. i.e. 25%, 50%, 75%, 100%
- Phase transitions are probably the best-known milestones. They are typically used for projects that concentrate on developing or expanding a product or service

Examples

- Installation, commissioning, or testing of equipment or facilities
- Rollouts of prototypes or samples
- Approval of patent applications
- Client training conducted
- Release of communications, such as announcements, press releases, and status reports
- Signings and negotiations of contracts

Estimate duration: It is NOT a guess

- Estimate the effort (actual time required for one person to complete the activity)
- Convert the effort to duration (or elapsed time) for each activity
- Determine how many people will be assigned to each activity
- Determine how many productive hours per day a person is available to work on the project

Techniques: Expert judgment, Analogous estimating, Parametric estimating, Three-point estimating, Reserve analysis.

The Gantt Chart: How to do it

Define the requirements can be helped by identifying the constraints on the project, which typically fall into three areas:

- Work, including tasks that must be completed and tasks that need not be completed. This may also include measures to determine the success of the project.
- Resource, including people available, machines, tools, materials, and general discretionary budget.
- Time, including the calendar time by when the project must be completed, holidays and other periods when resources are not available.

Identify all tasks that need to be completed to meet the requirements from step 1. Use the WBS and remember that each task should be uniquely identifiable.

Identify what is required to complete it, such as specific skills and resources.

Perform allocation of resources to tasks, considering the requirements from step 3 and the resources available.

Estimate the time that will be taken to complete each task, given the allocated resources

Identify task priorities and dependencies

Calculate the start and end dates for each task

Identify a scale for the Gantt Chart, for example one increment on the bar may represent anything from 1 hour to 1 month. Typically, this will enable the chart to be drawn within a single page width

Identify periods during which tasks may not be worked on, such as public holidays, private holidays, conferences, and company events which must be attended. Mark these on the chart.

Draw the horizontal bars on the Gantt Chart, against each task, using the information gained in steps 2 to 9.

Investigate how well the available resources are used by summing down the time taken in each column into a Histogram for each resource. This Histogram may then be used to perform load balancing by identifying actions to change tasks, resources, allocations, and times that will enable the resources to be optimally used. You may need to go back to step 5 and repeat this process until a satisfactory chart is obtained.

Regularly re-estimate outstanding work on tasks, taking note of the accuracy of previous estimates.

Critical Path

The Critical activities cannot be delayed without delaying the completion of the project.

Non-critical activities can be delayed without delaying the completion of the entire project

Risk Management

Risk IS NOT a problem, is the recognition of any uncertain factor that interferes on the project

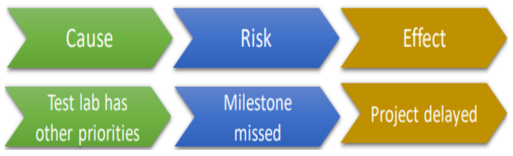
When estimating, there is the temptation to cover we by building up some extra time or cost.

Contingency: allowance of spare capacity, time, or money, built-in into the estimate to cover if something goes not according to plan.

Sources: Client, Design, Implementation, ...

Identify

Risk definition: An uncertain event or condition that, if it occurs, has a positive or a negative effect on a project objective. It includes both threats to the project objectives and opportunities to improve on those objectives.



Risk are not identified using the cause-risk-effect format. Risks identified are general rather than specific. Some things considered to be risks are not uncertain; they are facts and are therefore not risks.

Analyse

Give each risk identified a probability and impact it could have:

- Probability is the current estimate of the likelihood of occurrence
- Impact indicates the likely effect on project and the resulting product

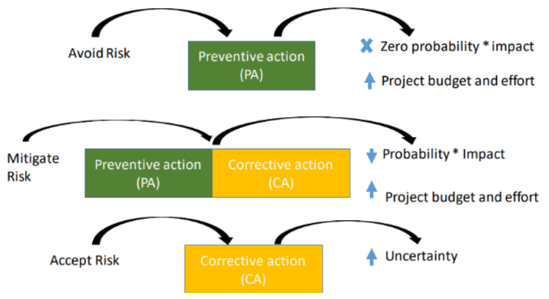
Risk	Consequence	Probability	Impact	Factor	Mgmt action
Neighbours complaints stop project	Airport cannot open	High (4)	Very high (5)	20	Define communication plan
Airlines nor interested	Airport cannot operate	Low (2)	Very high (5)	10	Early market study
Key team member leaves	Project delay	Medium (3)	Low (2)	6	Establish team induction protocol

Evaluate

- Avoid risk – change plan to eliminate the threat entirely
- Mitigate risk – reduce impact and/or likelihood
- Transfer risk – outsource or share the risk to a 3rd party and the ownership of the response (i.e. insurance policies, outsourcing specialist tasks)
- Accept risk – take the impact and budget the cost
- Escalate risk – threat is outside the scope of the project or exceed PM's authority.

Once we've decided the strategy for each risk is necessary to quantify the action cost. The costs related to mitigate, avoid, and transfer actions are preventive costs. In this case the cost become a direct project cost and the risk grade is modified. The costs related to accept actions (it also includes the part of the mitigated risks that we accept) are the contingency margin. In escalate risk is no longer monitored and if it happens, it is managed as a change, (no cost associated)

Preventive and Contingency Costs:



$$Contingency\ margin = \sum_{i=1}^{n} CA\ Cost_i \cdot CA\ Probability_i$$

Corrective Action (CA)	CA Cost	CA Probability	Contingency reserve
Buy damaged material again	5.000 €	20%	1.000€