* **What is Risk Analysis and Management:**
  + Series of steps to understand and manage uncertainty in software projects.
  + Identify potential problems (risks), assess their likelihood and impact, and establish contingency plans.
* **Who Does It:**
  + Involves everyone in the software process, including managers, software engineers, and customers.
* **Why Is It Important:**
  + Essential due to the complexity of software projects and the potential for things to go wrong.
  + Being prepared and managing risks is a key element of good software project management.
* **Steps in Risk Analysis and Management:**
  + **Risk Identification:** Recognizing potential risks.
  + **Risk Analysis:** Assessing the likelihood and impact of each risk.
  + **Risk Ranking:** Prioritizing risks based on probability and impact.
  + **Risk Management Plan:** Developing plans to manage high probability and high impact risks.
* **Work Product:**
  + Risk Mitigation, Monitoring, and Management (RMMM) Plan or a set of risk information sheets.
* **Ensuring It's Done Right:**
  + Risks should be derived from a thorough study of people, product, process, and project.
  + The RMMM should be revisited as the project progresses to keep risks up to date.
  + Contingency plans for risk management should be realistic.
* **Reactive Risk Strategies:**
  + Often referred to as the "Indiana Jones school of risk management."
  + Characterized by a lack of proactive planning.
  + Relies on reacting to problems as they occur.
  + Resources are set aside to address risks if they become actual issues.
  + Typically, nothing is done about risks until problems arise, leading to a "firefighting" approach.
  + When this approach fails, it may lead to "crisis management" and put the project in jeopardy.
* **Proactive Risk Strategies:**
  + Involves proactive planning that begins before technical work starts.
  + Identifies potential risks, assesses their probability and impact, and ranks them by importance.
  + Focuses on risk avoidance but also includes developing contingency plans to respond effectively to unavoidable risks.
  + A more intelligent and structured approach to risk management.
* **Definition of Software Risk:**
  + Software risk is characterized by two key attributes:
    1. **Uncertainty:** Risks may or may not occur, and there are no 100% guaranteed risks.
    2. **Loss:** If a risk materializes, it results in unwanted consequences or losses.
* **Types of Software Risks:**
  + **Project Risks:** These risks threaten the project plan, potentially causing schedule delays and increased costs. They encompass issues related to budget, schedule, personnel, resources, customers, and requirements.
  + **Technical Risks:** These risks threaten the quality and timeliness of the software being developed. They may lead to difficulties in implementation and include design, implementation, interface, verification, maintenance, and technical uncertainty issues.
  + **Business Risks:** These risks jeopardize the viability of the software project or product. They include risks such as building a product no one wants (market risk), not aligning with the company's business strategy (strategic risk), challenges in selling the product, changes in management focus (management risk), and budget or personnel commitment risks. Some risks may be unpredictable.
* **Categorization of Risks:**
  + Known Risks: Risks that can be identified through careful evaluation of the project plan, the business and technical environment, and reliable information sources.
  + Predictable Risks: Risks that can be extrapolated from past project experiences, such as staff turnover, communication issues with customers, and dilution of staff effort during ongoing maintenance.
  + Unpredictable Risks: Risks that are challenging to identify in advance but can and do occur unexpectedly.
* **Types of Risks:**
  + There are two types of risks in software projects: generic risks and product-specific risks.
  + Generic risks are potential threats to every software project, while product-specific risks are unique to a specific project and require a clear understanding of the project's context.
* **Risk Identification Methods:**
  + Risk identification involves creating a risk item checklist.
  + The checklist focuses on known and predictable risks in various categories, including product size, business impact, customer characteristics, process definition, development environment, technology to be built, and staff size and experience.
  + The checklist helps estimate the impact of risks and can be organized by answering questions relevant to each category.
* **Assessing Overall Project Risk:**
  + A set of questions helps assess the overall risk of a software project.
  + These questions include topics like commitment from top management, end-user enthusiasm, requirement understanding, project scope stability, team skills, and customer agreement.
  + The more negative responses to these questions, the higher the project's risk.
* **Risk Components and Drivers:**
  + The U.S. Air Force's approach involves identifying risk drivers that affect risk components, including performance, cost, support, and schedule.
  + Risk components are defined as performance risk, cost risk, support risk, and schedule risk.
  + The impact of each risk driver on a risk component is categorized as negligible, marginal, critical, or catastrophic, based on the potential consequences of errors or failure to achieve desired outcomes.

**Risk Projection:**

* Risk projection, also referred to as risk estimation, is a critical step in risk management. It involves evaluating each identified risk in two aspects: the likelihood or probability of the risk occurring and the potential consequences or impact if the risk does materialize.
* This process is a collaborative effort involving the project planner, other managers, and technical staff. The objective is to assess and prioritize risks effectively.

**Developing a Risk Table:**

* A risk table is a practical tool used to facilitate risk projection. It provides a structured format for documenting and assessing risks.
* In the risk table, each risk is listed along with its type or category. Common risk categories may include project size (PS), business (BU), customer (CU), technical (TE), and more.
* Risks are evaluated for their likelihood of occurrence (probability) and their potential impact on the project and product. Probability estimates are often obtained through discussions among team members.
* Impact values are categorized as 1 (catastrophic), 2 (critical), 3 (marginal), or 4 (negligible) based on predefined criteria.
* The risk table is sorted based on both probability and impact. High-probability, high-impact risks are given priority, while low-probability risks are deprioritized.

**Assessing Risk Impact:**

* Risk impact is influenced by three factors: the nature of the risk, its scope, and its timing.
* The nature of the risk refers to the potential problems that may arise if the risk occurs. For example, technical risks might involve issues related to interface compatibility, which can lead to integration problems.
* The scope of a risk combines severity (how serious it is) with its overall distribution (how much of the project or how many customers are affected). The larger the scope, the more significant the impact.
* The timing of a risk considers when and for how long the impact will be felt. In some cases, earlier exposure to a risk may be more favorable.

**Risk Assessment:**

* Risk assessment aims to further examine the accuracy of risk estimates and rank the identified risks.
* It involves defining a risk referent level, which represents the project's tolerance for risk. This level is often based on the critical risk components of the project, including performance, cost, support, and schedule.
* The risk referent level helps establish criteria for deciding when a project should be terminated due to the accumulation of risks that exceed the defined threshold.
* The process also considers compound combinations of risks and their potential effects on the project's referent levels.
* The assessment of risk is an iterative process, with ongoing review and updating of the risk table as the project progresses and new circumstances emerge.

**Refining Project Risks Using CTC Format:**

* During the early stages of project planning, risks may be stated in a general manner, often with a broad view of potential issues.
* As the project progresses and more information becomes available, it is possible to refine these general risks into more specific and manageable risks.
* One technique for refining risks is to use the Condition-Transition-Consequence (CTC) format. In this format, a risk is expressed as follows: "Given that <condition>, then there is concern that (possibly) <consequence>."
* The CTC format helps in breaking down a general risk into more specific conditions, transitions, and consequences, making it easier to analyze, monitor, and manage the risks effectively.

**Example Using CTC Format:**

* The book provides an example related to the reuse risk mentioned earlier (in Section 6.4.2) regarding the potential underutilization of reusable software components.
* The initial general risk stated that if all reusable components do not conform to design standards, there may be a concern about integrating only 70 percent of the planned reusable modules into the system, resulting in the need to custom engineer the remaining 30 percent of components.
* This general condition can be further refined into subconditions:
  1. Subcondition 1: Certain reusable components were developed by a third party with no knowledge of internal design standards.
  2. Subcondition 2: The design standard for component interfaces has not been solidified and may not conform to certain existing reusable components.
  3. Subcondition 3: Certain reusable components have been implemented in a language that is not supported on the target environment.
* The consequences associated with these refined subconditions remain the same, which is that 30 percent of software components must be custom engineered. However, the refinement helps to isolate the underlying risks, making it easier to analyze and respond to each specific risk condition.

**Risk Mitigation, Monitoring, and Management:**

* The primary goal of all risk analysis activities is to assist the project team in developing an effective strategy for dealing with risks.
* An effective risk management strategy considers three key aspects: risk avoidance, risk monitoring, and risk management and contingency planning.
* Risk avoidance is the preferred strategy, which is achieved through risk mitigation. This involves taking proactive steps to reduce the likelihood and impact of identified risks.
* An example is given of the risk of high staff turnover, where potential mitigation strategies include understanding the causes of turnover, addressing them before the project starts, and planning for continuity if turnover occurs during the project.
* As the project progresses, risk monitoring activities are initiated. Project managers closely observe factors that may indicate whether a risk is becoming more or less likely. For the staff turnover risk, factors like team morale, interpersonal relationships, and job opportunities are monitored.
* The effectiveness of risk mitigation steps is also monitored, such as ensuring that documentation is thorough and can stand alone, aiding in knowledge transfer if needed.
* Risk management and contingency planning are necessary if mitigation efforts fail and a risk becomes a reality. In the case of high staff turnover, contingency planning may involve temporarily reallocating resources and capturing knowledge before team members depart.
* It's important to note that implementing risk mitigation and contingency planning steps incurs additional project cost. Therefore, a cost-benefit analysis is essential to evaluate whether the benefits of these steps outweigh the associated costs.
* The planner should decide which risks are critical and require detailed risk management plans. The Pareto principle (80-20 rule) is applied to software risk, where 80 percent of the overall project risk can be attributed to 20 percent of the identified risks.
* Not all identified risks may make it into the Risk Mitigation, Monitoring, and Management (RMMM) plan; they are included based on their potential impact on the project's success.

In summary, this section emphasizes the importance of proactive risk management in software projects, including the development of mitigation strategies, continuous monitoring of risk factors, and the establishment of contingency plans when risks become realities. The decision to include specific risks in the RMMM plan is based on their significance in terms of project success.

**Safety Risks and Hazards:**

* Risks in software projects are not limited to the development phase but can also occur after the software has been successfully delivered to the customer. These post-delivery risks are often related to the consequences of software failure in real-world applications.
* Historically, there was hesitation to use computers and software to control safety-critical processes, such as nuclear reactors, aircraft flight control, weapons systems, and industrial processes, due to concerns about the consequences of failure. A single undetected fault in a computer-based control or monitoring system could lead to severe economic damage or even human injury or loss of life.
* However, the advantages of computer-based control and monitoring, including cost-effectiveness and enhanced functionality, have led to the widespread use of software in safety-critical systems today.
* When software is part of a control system, the complexity can increase significantly. Subtle design faults induced by human error, which can be detected and corrected in conventional hardware-based control systems, become more challenging to identify and eliminate in software-based systems.
* Software safety and hazard analysis is a branch of software quality assurance that focuses on identifying and assessing potential hazards that could negatively affect software performance and lead to system failures.
* Early identification of hazards in the software engineering process allows for the specification of design features that can either eliminate or control these potential hazards, reducing the risk of software failure in safety-critical systems.

In summary, this section emphasizes the importance of analyzing and mitigating safety risks and hazards associated with software in safety-critical applications. By identifying and addressing these hazards early in the software engineering process, it is possible to design features that enhance safety and reduce the potential for system failures.

**The RMMM Plan:**

* A risk management strategy can be integrated into the software project plan, or the risk management steps can be organized into a separate Risk Mitigation, Monitoring, and Management Plan (RMMM Plan). The RMMM plan documents all the work performed during the risk analysis process and is a crucial part of the overall project plan.
* While some software teams include risk management within the main project plan, others prefer to create a separate RMMM document. In the latter case, each risk is individually documented using a risk information sheet (RIS). These RIS documents are often managed in a database system, allowing for easy creation, data entry, priority ordering, searches, and analysis.
* The RIS format typically includes information about the risk identification, likelihood, impact, risk exposure, risk mitigation steps, and risk monitoring procedures. It provides a structured way to capture and manage information about each identified risk.

**Risk Mitigation and Monitoring:**

* Once the RMMM has been documented and the project is underway, risk mitigation and monitoring activities begin.
* Risk mitigation focuses on taking proactive steps to avoid potential problems related to identified risks. This includes measures such as identifying causes of the risk, taking steps to mitigate these causes, and developing strategies to ensure project continuity in case the risk materializes.
* Risk monitoring is a tracking activity with several objectives:
  1. To assess whether the predicted risks are actually occurring during the project.
  2. To ensure that the risk aversion steps defined for each risk are being correctly applied.
  3. To collect data and information that can be used for future risk analysis.
* Risk monitoring is also essential for identifying the origins of problems during the project. It helps trace which risk or combination of risks might have contributed to issues as they arise.
* Effective risk monitoring allows project managers to make informed decisions and adjustments to the project plan, helping to keep the project on track and minimize the impact of potential risks.

In summary, this section underscores the importance of having a well-documented RMMM Plan as part of the software project management process. It outlines the key components of the RIS format and emphasizes the significance of risk mitigation and monitoring activities for managing and mitigating potential risks during a project.

See that summary and diagram from the book