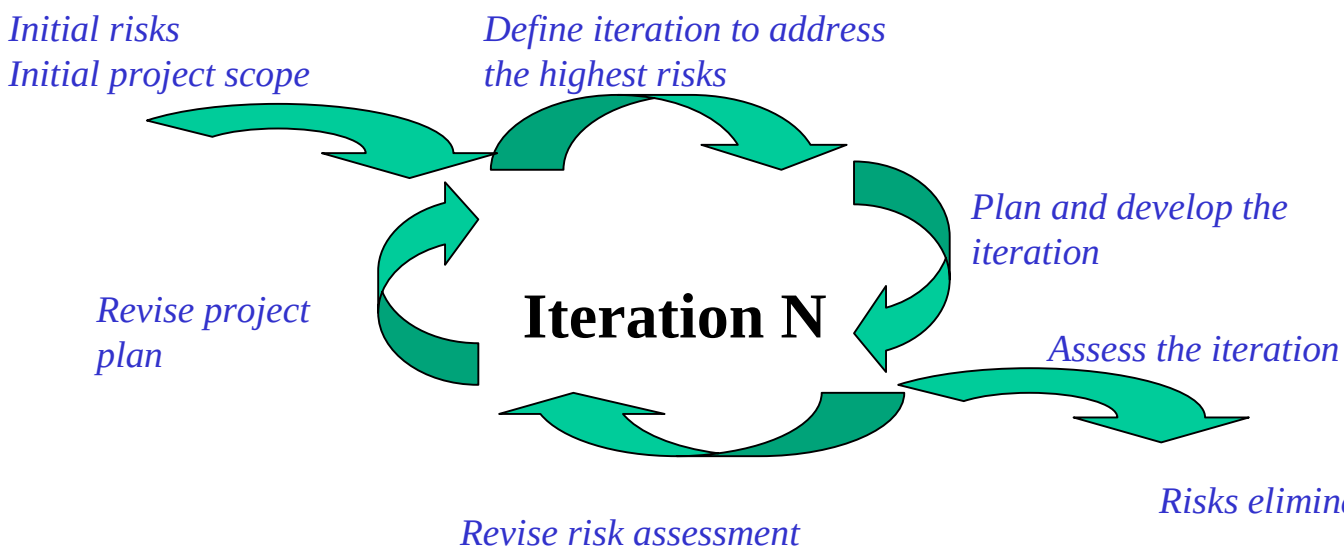
Design Planning

1. **Rationale**
2. **Iteration Planning**
3. **Risk Analysis**
4. **Quality Planning**
5. **Example**



1. Introduction

Rationale

☞The architecture team plays the same role as the military staff leading a military campaign: the ***failure or the success of the whole campaign depends on the strategies and decisions they make***.

☞A necessary condition for the success of a project: ***identification*** of all the ***potential sources of failure***, broadly referred to as the ***risks*** for a project.

☞A project may fail because for instance:

- the ***resources and time*** needed have been ***underestimated***,
- the ***available staff*** doesn't have the ***necessary skills***,
- there is a ***shortage of skilled personnel***,
- skilled ***personnel leave*** the project before end,
- the appropriate ***technology is not available***,
- the ***functionality*** required has ***not been properly identified*** or ***implemented***.

☞ It is the ***responsibility of the architecture team*** to identify the architectural risks involved in the project, and to ensure that they are under control, and will be mitigated.

☞**Design planning** can be defined as a range of guidelines and strategies defined by the architecture team that guides the system construction.

- ***Main goal of the design planning process*** is to identify the design issues and the risks underlying the construction of the system, and to find appropriate strategies in order to manage them.

- It is recommended to (identify and) focus only on a limited number of risks (***the most important ones: 5-15***).
  - This requires identification of all the risks
  - Then prioritize them in order to select the most important ones

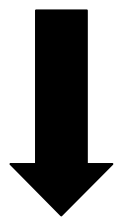
☞Design planning includes also the definition of ***quality*** requirements and the organization of the tasks involved in achieving them.

Motivating Example

The company in charge of the development of the Hospital Information System has charged one of its subcontractors of the development of the ***Patient Monitoring*** *subsystem, which is an essential part of the system. Due to high competition,* ***short time-to-market*** *is essential.*

The subcontractor is a small software development company (12 developers) specialized in mission critical software development , which has established a strong reputation for ***meeting deadlines****. Because of the mission critical nature of its products, the subcontractor* ***prefers building*** *most of its components in-house, rather than buying.*

- Constraints Gathering
- Prioritize Constraints



Aggressive schedule!!!

2. Iteration Planning

-RUP is an iterative, risk-driven, and use-case centric process.

- Iterations can be planned around architecturally significant use cases by following a priority list.
- Use cases can be prioritized according to the significance of the risks involved.

Architecturally significant use cases

-Are the ones that cover the main tasks or functions the system is to accomplish and/or could possibly impact the architecture

-Include:

- ***Critical use cases***: those that are most important to the users of the system
- Use cases that ***carry significant software risks***.
- Use cases that ***carry quality requirements***.

-Secondary and optional use cases are not key to the architecture.

Use Case Description

-In addition to the graphical and textual description of use cases, a tabular description add to clarity and decision making, and greatly helps in change management and establishing requirements traceability.

- *Template/Example*

ID	Use Case	Actors	Prio-rity	Quality Constraints		Risks
				Attribute	**Parame-ter**	
UC_001	Login to system	Patient, Doctor, Nurse	2	Performance (Response Time)	<3sec	Fail to meet security standard
UC_002	Patient_monitoring	Nurse, Doctor	3	Reliability	>99.99999	Lack robustness

3. Risk Analysis

Cardinal Aims of a Project

-The success of a project can be measured by the degree of achievement of its *cardinal aims*, which are of three kinds:

- the ***whole life costs*** of the system: development and ownership costs
- the ***system goals***: main system's objectives from the business or the customer standpoint.
- the ***side effects***: unexpected ***beneficial*** or ***detrimental*** characteristics or capabilities of the systems appearing during the development.

Notion of Risk

-Any threat that can lead to the failure of a project.

-Any threat to the achievement of the cardinal aims of a project.

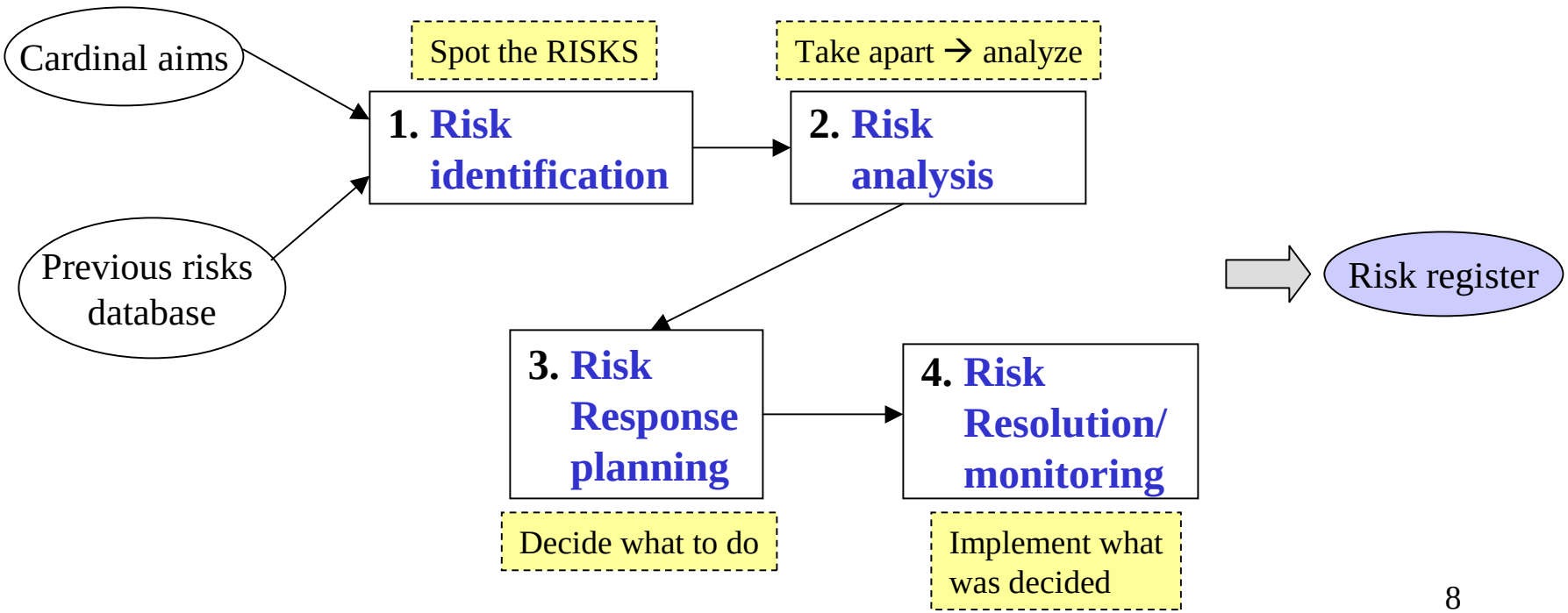
-*Examples:*

- ***Misunderstood requirements*** -> need to rework the sw. after delivery
- ***Staff new to the language*** -> slow development, poorer code delivered
- ***Weak User Interface*** -> customer dissatisfaction etc.

Risk Management Process

-A risk management process involves the following steps:

1. Risk **identification**
2. Risk **analysis**
3. Risk **response planning**
4. Risk **resolution and monitoring**



Risk Identification

-A risk always involves two aspects: a ***cause*** and an ***effect***.

- The ***cause of a risk can be another risk***, and its effect can induce new risks.
- We can attach a ***probability*** to the cause, and an ***impact*** or a ***size*** to the effect.

-A starting point for risk finding can be the identification of the project ***cardinal aims***.

- Then by proceeding backward we can identify any potential causes that may threaten their achievement.
- We build that way a ***cause-effect tree*** in which the roots are direct threats to the cardinal aims, and the leaves are basic risk factors.

-The risks identified in the tree are numbered, and recorded in a ***risk register***, with a short description and their direct effect.

- The risk register is ***built and updated progressively*** as more risks or more information on existing risks arrive.

Recommendation:

- Brainstorm with team to develop list of RISKS
- Review and question each risk for relevancy
- Don't waste resources on things that won't happen
- Obtain input from staff to update the Risks (workshops)

Possible Sources of Risk:

- Vague or changing requirements
- Failed Product Performance
- Lack of standards and codes of practice
- Volatile or innovative environment
- Difficulty of time and resource estimation
- Reliance on scarce skills
- Reliance on suppliers

Example: Warehouse Management System

We are delivering an automated system that will run a big warehouse that is being upgraded to improve its accuracy and increase its capacity by doubling its throughput.

The system will:
-accept new stock and move it to its appointed storage point
-print out the day's 'pick-lists' for smaller items so that staff can place them in bins destined for stores
-move these bins and the larger items to the loading bays at the right moment for the waiting vehicles which will deliver them to the branches.

We have chosen the warehouse management product that we will buy as basis of our system. It will need some customization for the warehouse and product lines. The warehouse will need to be converted, staff trained in the new prices, the system installed.

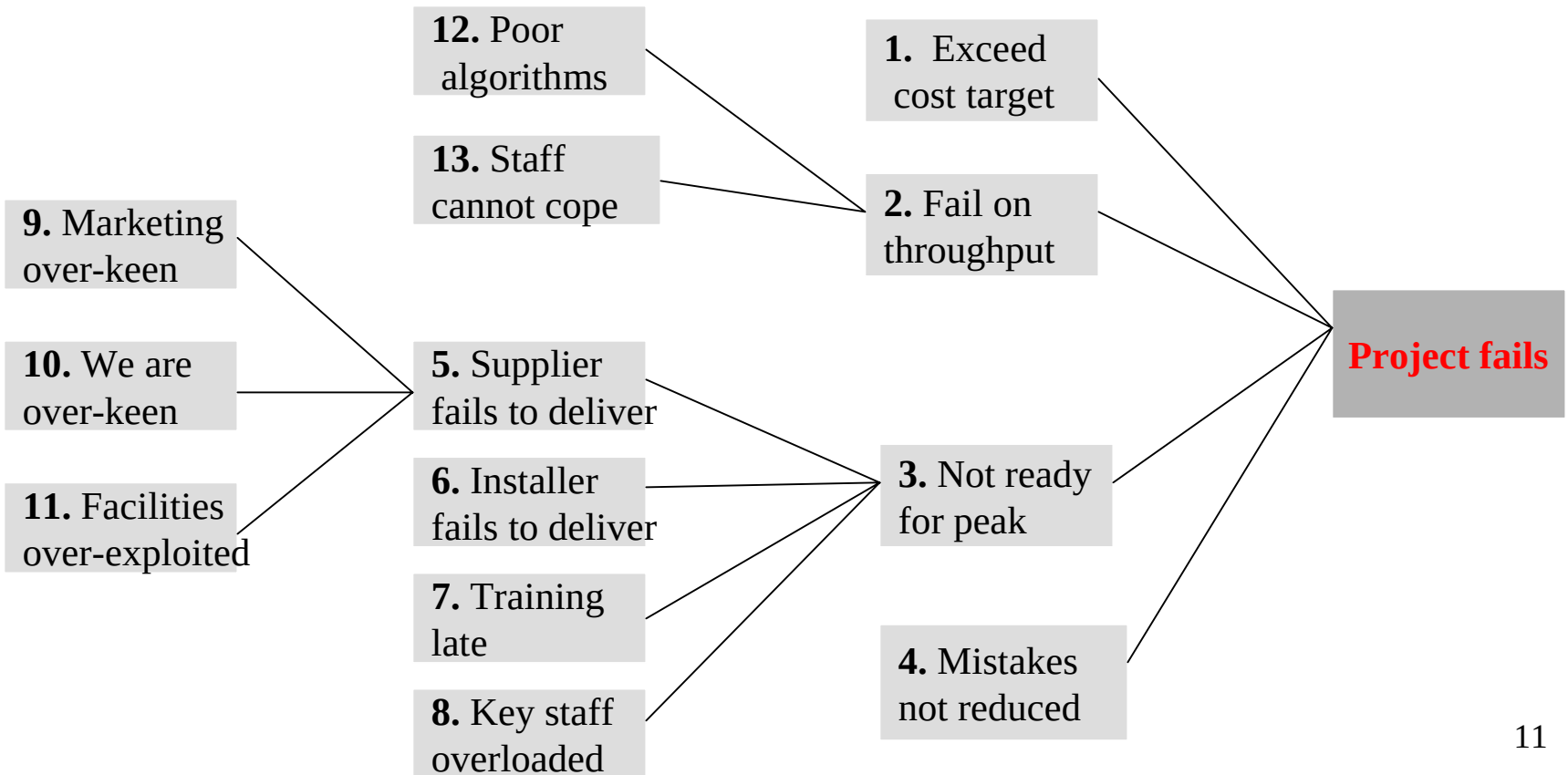
It is expected that the system will be ready for the next peak period and cost no more than $10 million.

Cardinal Aims of the Project

-Product delivery at a cost of $10million
-Double throughput capability at the times of peak demand
-Reduce number of mistakes in what is sent to branches to a third of its current value
-Be ready for the start of the peak demand next year.

Focus on finding the RISKS, then deal with them

Cause-effect Tree



Risk Register

-Each risk is assigned a ***unique identifier***, and is allocated to a specific ***person in charge*** of its ***management and monitoring***.

Risk	**Description**	**Effect**	**Source of Uncertainty**	**Nature**	**Probability**	**Impact**	**Chosen Risk Reduction Measures**
<A unique risk identifier>	*<A brief description of the risk in cause-effect terms>*	*<A list of the risks that this one itself causes>*	*<An indicator saying whether the risk is caused by event and/or estimating uncertainty>*	*<A description of the event and/or estimating uncertainty that is causing the risk>*	*<An assessment of the likelihood that the risk materialize>*	*<An assessment of the scale of the impact the risk could have if it materialized >*	*<A list of the pre-emptive and/or reactive measures chosen to manage the risk>*

The risk register lists the risks and their cause-effect relationships.

Example: Risk Register for the Warehouse Management System

Risk	Description	Effect
1	We exceed the development cost target	
2	At peak demand we cannot handle twice the throughput	
3	We are not ready for the start of the peak period next year	
4	We reduce mistakes but not by two-thirds	
5	The supplier of the warehouse management product fails to deliver the customized version in time for us to complete testing before the peak period	3
6	The machinery installer fails to get the machinery in place ready for integration with the software in time	3
7	We fail to get sufficient staff trained in the operation of the system in time	3
8	Key people who will be needed to push through the work are overloaded	3
9	The requirements being pushed by the marketing team are not fully validated	5
10	We demand more ‘knobs and whistles’ that can reasonably be incorporated in the time available	5
11	Potential new facilities are being built into the machinery and we feel obliged to exploit these from day one	5
12	The algorithms that control the movement and picking of stock may not be right for the way the warehouse is to be set up	2
13	Staff cannot cope with the new technology	2

Risk Analysis

-In order to conduct a thorough analysis of a risk, we need to have a good understanding of its nature, more specifically its ***cause*** and ***effect***.

-According to the ***impact*** of a risk, we can classify it either as a ***binary risk*** or a ***sliding risk***.

Binary risk: *a risk, which either fully materializes or not at all; there is no middle ground, either it happens or it doesn't happen at all.*

Sliding risk: *a risk whose impact can be variable; we may feel it slightly, hardly, or not at all, or somewhere in between.*

When discussing the impact of a risk, we are talking about its ultimate impact on the cardinal aims of the project.

-The cause of a risk may also be classified in one of two ways: ***event uncertainty*** or ***estimating uncertainty***.

Event uncertainty: *situation created or caused by an event that might or might not happen; the chances for it to happen are characterized by a* ***strong variability****.*

Estimating uncertainty: *a lack of concrete knowledge or a total ignorance about some facts or the occurrence of some events.*

Try to phrase the nature of an event uncertainty by forming a sentence that starts ***"It may happen that…"***

Try to phrase the nature of an estimating uncertainty by forming a sentence that starts ***"We are uncertain how much…"***

-An important aspect of the cardinal aims is that they are ***not always quantifiable***; in some cases the single metrics appropriate or usable are qualitative ones.

-A way of estimating the cause and effect of a risk, consists of using some ***qualitative ranking***.

- The ***probability*** attached to the ***cause*** of a risk can be ranked as:

> ***VL***: *Very Likely*
> ***L***: *Likely*
> ***U***: *Unlikely*
> ***VU***: *Very Unlikely*

- The ***impact*** associated to the ***effect*** of a risk can assessed as:

> ***L***: *Life threatening*
> ***P***: *Project threatening*
> ***E***: *Expensive in cost or time*
> ***S***: *Some cost or time penalty*
> ***N***: *Negligible cost or time penalty*

Risk Response Planning

-Identification of the risk reduction measures and strategies.

- Goal: ***remove the threats*** to the achievement of the cardinal aims of the project.

-Risk reduction may consist of ***removing the cause*** of the risk by reducing its ***probability*** or by alleviating its ***impact*** on the project, or a combination of both approaches.

-Risk reduction measures are classified in two categories: ***pre-emptive measures*** and ***reactive measures***.

Pre-emptive measure: *is executed before the risk materializes; its focus is mainly the reduction of the probability associated to the cause of the risk.*

Reactive measure: *is executed after the risk materializes; its main goal is to minimize the impact of the risk.*

-***Pre-emptive measures*** are divided into four sub-categories: *information-buying*, *risk-influencing*, and *contractual-transfer* measures and *process models*.

Information-buying**:** *collects information or knowledge that will dissipate the uncertainty underlying the cause of the risk; so it is mainly used for risks based on* ***estimating uncertainty****.*

prototyping, modeling, research, measurement

Risk-influencing: *targets the reduction of the variability attached to the cause of a risk probability; is mainly used for risk based on* ***event uncertainty****.*

monitoring, training

Contractual transfer: *consists of subcontracting risky activities to a third party that is specialized in such kind of activities.*

transfer risk to specialists

Process model: *structures the project development into steps that are risk-driven, and allows to deal with major risks through successive and iterative steps.*

ordered tasks to reduce possibilities

-***Reactive risk reduction measures*** consist of two sub-categories: *contingency plans* and *insurance*.

Contingency plan: *backup or alternative solution that will be adopted in case where the risk materializes.*

Insurance: *consists of getting coverage in case where the risk materializes.*

-***Residual risks***: risks remaining unsolved after applying the adopted measures.

- We may provide an assessment of a residual risk using a 3-level scale: ***worst case***, ***best case***, and **chosen case**.

- The chosen case lies between the best and worst case, and represents the ***risk provision***.

Risk	Description	Effect	Source of Uncertainty	Nature	Probability	Impact	Reduction Measures
6	The machinery installer fails to get the machinery in place ready for integration with the software in time	3	event	It may happen that the installer does not have the capability to deal with our requirements and to do the job	U	P	Ensure the supplier understand the business criticality for both parties. Increase the visibility of the project in the industry
7	We fail to get sufficient staff trained in the operation of the system in time	3	estimate and event	We are uncertain how long it will take to train them. It may happen that we cannot get sufficient candidates to train	L	E	Get some estimates from a training co. for the training requirements for the tasks concerned. Find out how long things took at the last major change and scale the figures up. Increase wage levels for trained staff to improve recruitment
8	Key people who will be needed to push through the work are overloaded	3	estimate	We are uncertain how big the load on them will be and what other commitments they will have at the time	L	S	Inventory other activities currently occupying time of key staff. Negotiate with relevant directors to free more of their time

4. Quality Planning

Notion of Software Quality

***Quality**: the totality of features and characteristics of a product, process or service that bear on its ability to satisfy stated or implied needs.*

-The expected quality features and characteristics of a software product are commonly referred to as ***Quality Attributes***.

- Software quality attributes are assessed by using, when possible, appropriate software measurements or metrics.

- Quality attributes are ***used early in the development process*** to identify user quality requirements. Each system has specific and unique quality needs, which are a function of the purpose of the application.

- Quality factors for a given system ***may be conflicting***; for instance achieving security may be at the expense of performance or interoperability.

-There are two main categories of quality attributes: functional (what a product should do) and non-functional (what a product should be like)

Functional quality attributes: apply to pieces of software, from the smallest components to entire systems.

Examples:

- *'when the pressure sensor reading climbs through a pressure level of 3.2 bar, the relief valve control line shall be set to open'*
- *'all relevant data shall be secured to disc before any transaction is cleared'*

Non-functional quality attributes: apply to any product of the development process : specifications, code, manuals, or final system.

Examples:

- *'the system shall be capable of operation on a computer with 64Mb of memory'*
- *'the system shall give uninterrupted service despite any power outages of up to 2s duration'*

Typical (non-functional) Quality Attributes

Life cycle	**Product operations**	**Product revision**	**Product transition**
Quality factors	Correctness Reliability Efficiency Usability Integrity Safety Security	Maintainability Flexibility Testability	Portability Reusability Interopera-bility

-***Correctness*:** conformance of a program to its specification

-***Reliability***: ability of a program to achieve precisely its intended mission.

-***Efficiency*:** amount of computing resource required by a program to achieve a given task.

-***Usability*:** effort required to learn, operate and use a program

Typical Quality Attributes (ctd.)

-***Maintainability*:** effort required modifying, updating, evolving, or repairing a program during its operation.

-***Flexibility*:** effort required to evolve or modify an operational program.

-***Testability*:** effort required to test a program.

-***Portability*:** effort required to transfer a program from one computing environment or platform to another.

-***Reusability*:** ability and effort required to reuse a program or part of a program in other applications.

-***Interoperability*:** effort required interconnecting or relating two different applications, running possibly in different computing environment.

Common Design Principles/Strategies

-Used for solving concrete design issues or risks.

- Used to build proactively quality in software products
- Famous design principles were proposed and are widely used for key quality attributes such as maintainability, security, reliability, and efficiency.

Example: Design Principles for Maintainability

Principles	**Role**	**Application**
Modularization	Ease reuse and modifiability	High cohesion and low coupling.
Separation of concerns	Ease reuse and modifiability	Separate unrelated concerns and group similar ones.
Separation of policy and implementation	Ease reuse and modifiability	Decouple policy or rules from their execution or enforcement.
Separation of interface and implementation	Ease modifiability.	Encapsulation, information hiding

Quality Plan

Definition: a document setting out the specific quality practices, resources and activities relevant to particular product, process, service, contract or project.

-Quality planning involves three activities:

- **Quality Achievement:** how we build the quality in.
- **Quality Control:** how we check that quality has been built.
- **Quality Preservation:** how we make sure that the quality stays in, despite any changes we make.

1. Quality Achievement Plan

-Characterization of system to be developed
-Client expectation or requirements on development
-Chosen development methods
-Chosen tool support
-Chosen target environment

Quality is achieved by using the right tool for the right job (how we build quality in)

1. Define the attribute(s)
2. Define the attribute check procedure
3. Carry out the check procedure
4. Record the result
5. Take and record any corrective action

2. Quality Control Plan

-Planned product types
-Specifications and standards
-Quality control activities

3. Quality Preservation Plan

-Identification control
-Change control
-Configuration control

Ensure that the quality is not damaged by subsequent changes

Example:

Quality Attributes

Quality Control Activities

'when the pressure sensor reading climbs through a pressure level of 3.2 bar, the relief valve control line shall be set to open'

Analyze the software connecting sensor reading to the valve control and verify this behavior will be its effect.

Set up a situation where the pressure climbs through 3.2 bar and observe the relief valve control line setting.

- Analyze and validate Vs. Observation
- How much can we realistically check?

'All relevant data shall be secured to disc before any transaction is cleared'

Analyze the design, in particular checking the locations of data updates at the point where every transaction is cleared.

5. Example

-Identify use cases, quality attributes and risks for the ATM System.

The bank's motivation for developing the system is to attract new customers by offering ***low banking fees****, and a variety of services. The bank will also be able to reduce its wage costs by processing an increased number of banking transactions automatically through the system instead of manually through cashiers. It is essential for them to lower the development cost and to minimize the product support cost.*

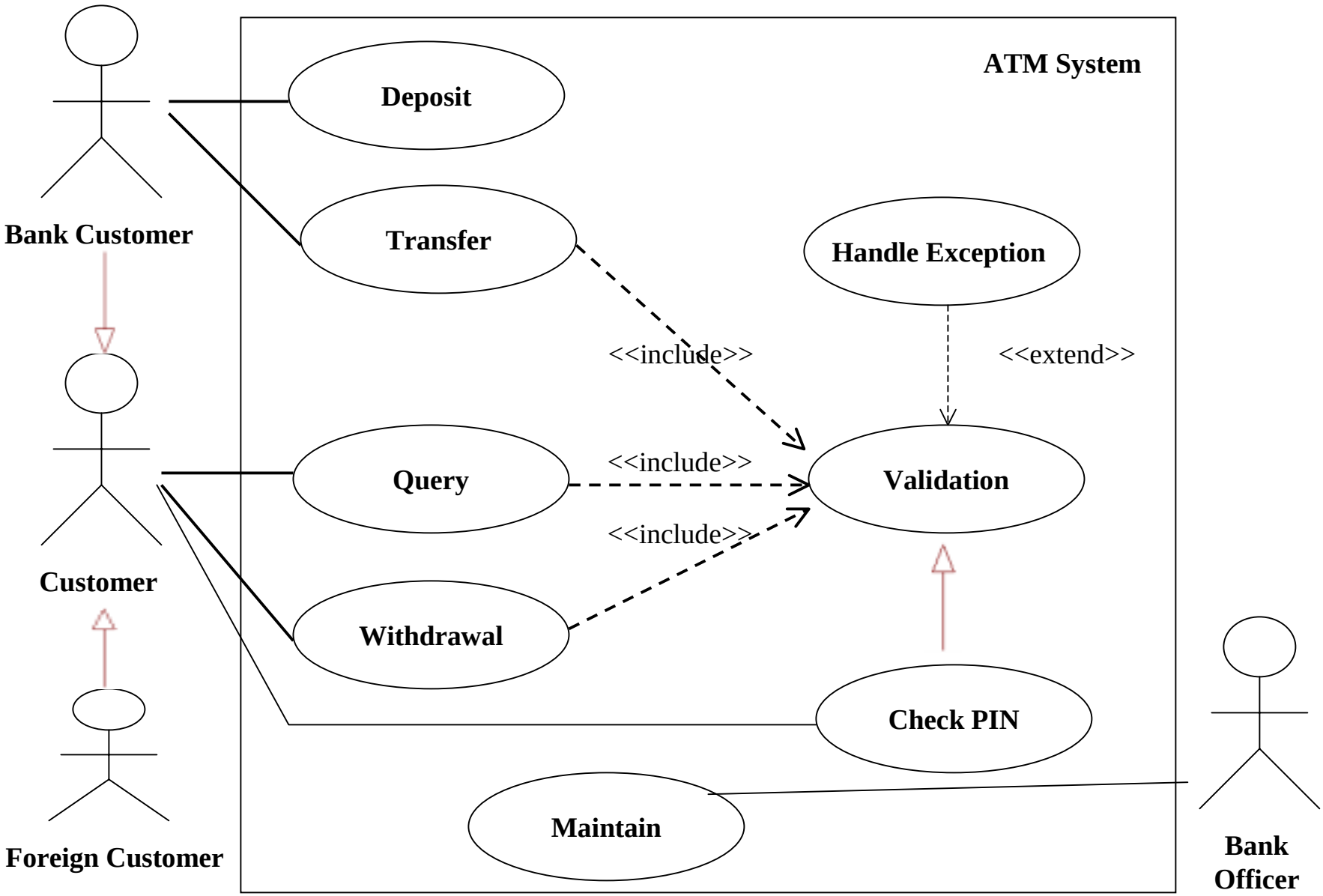
Security is critical; the system must be fully integrated into existing enterprise security infrastructure. More specifically the ATM system will reuse an existing secured database.

The time for 90% of the users to learn (through supplied step-by-step instructions) how to use the first time the system must not be more than 5 minutes.

When a user issues a request, the system should respond with a verification of the request within 1.0 second in 90% of the cases. The time for the verification must never exceed 10.0 s, unless the network connection is broken (in which case the user should be notified). The ATM System must have no more than 1 hour per month of down time.

Interbank Software Ltd. is a small software company which is known for its ability to meet deadlines and for its past experience in high integrity software development. There are ten developers working permanently for the company with solid background in object-oriented design and programming using UML, Java, and C++.

Use case diagram



Preliminary Analysis

Cardinal aims

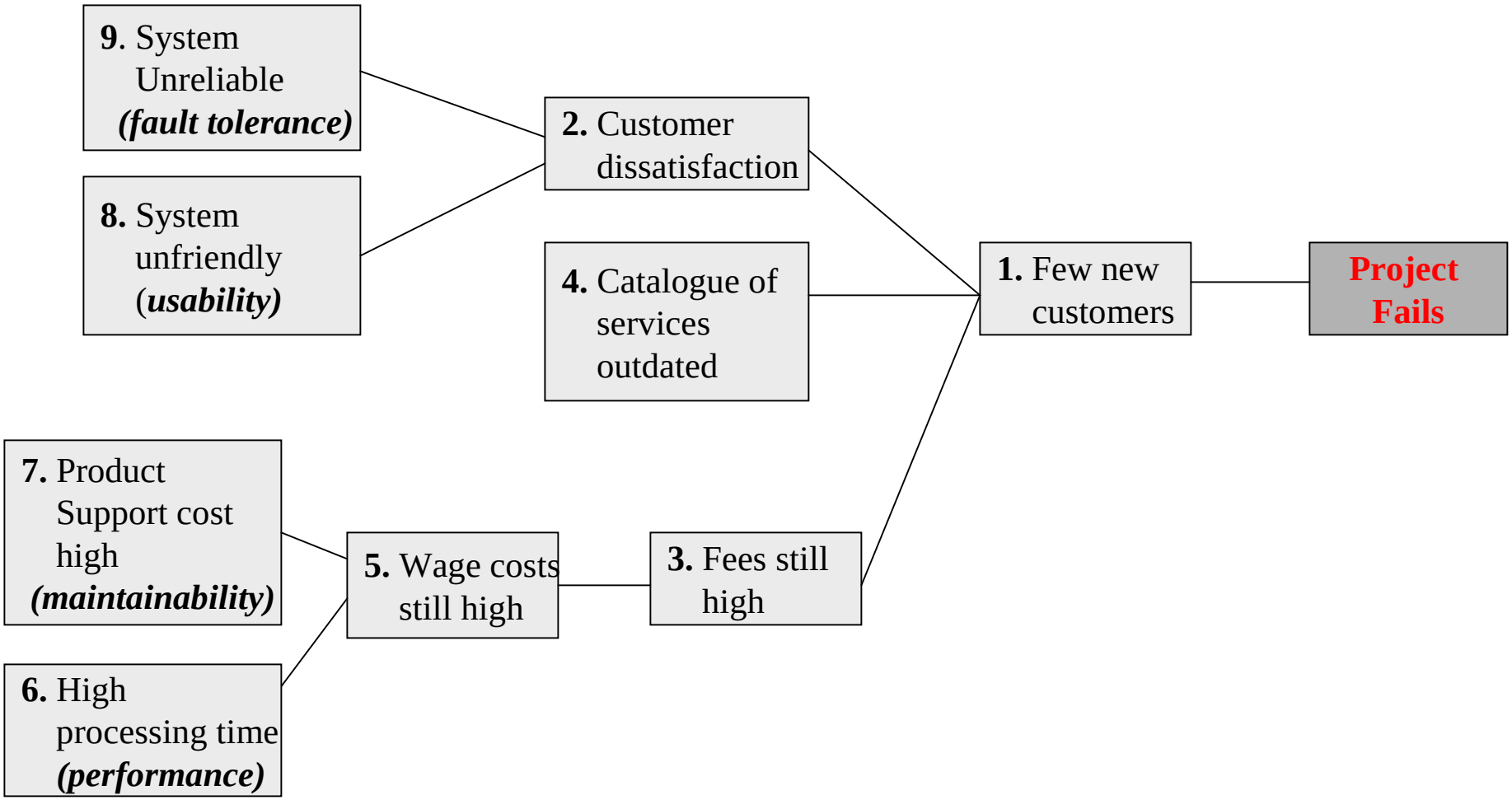
-Bank goals:

- attract new customers,
- provide low banking fees
- reduce processing fees, wage costs
- reduce development and product support cost
- provide a variety of services

Quality Attributes

- **Security:** reuse of existing security infrastructure
- **Usability:** 90% of users must learn how to use the system in <5min
- **Performance:** response time <1s for 90% of request
  verification time <10s
- **Fault tolerance:** detection, reporting
- **Maintainability:** low product support cost
- **Reliability:** Down time < 1h per month

Cause-Effect Tree



Risk Registry (samples)

Risk	Description	Effect	Source of Uncertainty	Nature	Probability	Impact	Reduction Measures
6	The system is very slow: minimum response time requirements are not met.	5	event	Hardware and network capacity are not enough to sustain the workload	L	E	Conduct some capacity planning during the design in order to guide and evaluate the workload distribution among modules. Validate performance metrics through testing.
7	Product may be difficult to maintain, which increases support cost.	5	event	Product structure lacks flexibility and is very complex.	L	E	Apply modular design principles:separation of concerns, separation of implementation and policy, separation of interface and implementation etc.
9	System may fail quite often.	2	event	Hardware and software unreliable.	U	P	Apply fault tolerant design principles and techniques (Modular Redundancy etc.) . Conduct some reliability and fault analysis during design. Validate reliability metrics through testing