Risk Management Plan

By Group2

5.2018

Catalogue

Introduction

Background of the Project

Sunstate was founded in 1977. The company's main business is to provide rental equipment and tools for construction, industry companies and so on, and provide reliable services for these companies to help them accomplish their tasks efficiently and safely. The company's earliest base was in Arizona. Rental equipment ranges from handmade equipment to heavy-weight equipment, with sets of complete construction equipment readily available. Of course, the equipment is maintained well, and the quality is guaranteed. With the expansion of the rental business, equipment leasing points are currently available in California, Nevada, Utah, Colorado, Arizona, New Mexico, Oklahoma, Texas, and Tennessee.

Due to the increase of leasing business, the traditional manual way of recording business has not been able to meet the needs, so a complete set of online leasing software system is needed to be created to record contracts, do inventory management, query business data, do financial statistics, write reports and other transactions for efficient leasing.

Purpose of the Document

The online rental system designed by our project is based on B2B pattern, which can provide online rental services for company level customers. This system will help customers to facilitate online leasing, select the equipment required and make a reservation time, and greatly improve the leasing efficiency. At the same time, the system will help Sunstate company to count customer rental information, inventory information, financial information and related information, reduce the error generated by manual records, and help Sunstate company carry out data analysis to improve service quality and profitability. In addition, the system will intelligently calculate the optimal allocation path based on real-time customer rental information, which can save a lot of manpower and improve the efficiency of equipment delivery.

What's more, we will do some necessary work first, so as to improve the efficiency of project development, save money and time. In recent years, there has been great progress in software development technology and tools, but the problem of software project development, overspending, or even not meeting the needs of users has not been fundamentally solved. There are many uncertain factors in the software project, which will affect the successful completion of the project, so we need to go into the management of these risks.

In the risk management of the project, we aim to decrease some risks to a certain extent, and we can have a certain coping strategy to reduce the loss caused by the risks hard to avoid. In a company which is likely to have a large number of users like Sunstate, there may be a lot of economic loss in practice if the projects have a lack of concurrent performance after publishing, and the risk management we do is able to avoid it to a large extent through some strategies.

Through steps of risk identification and risk assessment for the development process of the online leasing system of Sunstate company, this document gives some important risk coping strategies, and gives the implementation scheme of risk monitoring. It can effectively guide and assist the development process of software, and better manage the software project management for the item managers. It provides a valuable reference for identifying and preventing risk in advance; for system analysis designers, software developers and testers and database managers, by reading our risk management documents ahead of time, it is also possible to consciously circumvent the possible risk operations.

Scope of the Document

The online system provides services for leased customers to log in, create lease contracts, and also provide the Sunstate company with warehouse management, contract data query, task assignment, financial statistics, and the function of generating reports.

In order to extract important risks related to the development of these functions and to deal with risks, we have done the following works.

Risk identification

We use fish bone map, SWOT and other tools to assist us in risk identification. After group members brainstorming, we list more than 30 risks first, then vote for 15 higher priority risks, and then classify the selected risks from the cause of origin with RBS and make further amendments to ensure that the risk is identified comprehensively, Specific and accurate.

Risk assessment

In the qualitative analysis, we use the Delphi method to divide the criteria of risk probability and risk impact into 10 grades. Then each group member independently scored the probability and impact of the 15 risks selected in the risk identification, and uses the product of the average of two factors to determine the final risk severity level The probability influence matrix is made, and the results of qualitative analysis are presented more intuitively. In the quantitative analysis, we use the decision tree to evaluate the economic benefits that can be obtained after a certain response to these risks. At the same time, the sensitivity analysis method is used to find the sensitive factors in the development process of Sunstate online leasing system, and the influence degree of each factor is determined on the basis of the control variables, resulting in a tornado chart.

Risk response

Referring to historical cases, consulting information and combining the knowledge and experience of software engineering, software testing, and other disciplines, we give the coping strategies for the 5 highest priority risks selected in the risk assessment process. Consider why , how, risk, and specific measures (What), risk response for team members (Who), time to take risk response (When), and the money and time spent on risk response schemes, respectively. These aspects (How much).

Risk monitoring

We consider risk causes through causal analysis, so as to identify and measure project risks early in the process of risk monitoring. Then, referring to the audit review method and risk monitoring method, a series of tables were drawn as a regular recording tool for risk monitoring. As risks are not all avoidable, we will record the experiences and lessons learned from the risks that have taken place and caused loss. In order to better adapt to the development and change of the situation, we will continue to make feedback on the existing risk management work in the process of risk monitoring in order to adjust the strategy in time. Finally, some emergency measures for sudden risks are given.

Risk Identification

Input & Output

Risk Identification Outline

Risk Identification Process

Input & Output

Input

- Risk Management Plan
- Project Management Plan
- Description of project scope
- Corporate environmental factors
- Organization process assets
- Project Charter

Output

- Risk record
- Risk identified
- Possible responses
- Root cause
- Project Management Plan Update

Risk Identification Outline

Risk identification is the first step in risk management and the basis of risk management. Only on the basis of correctly identifying the risks that are faced, can we actively choose appropriate and effective methods for processing. The project risk identification means that the project undertaker, based on the collected data and investigation and research, uses various methods to systematically classify and fully identify the potential risks that have not yet occurred and the various risks that are objectively present. The risk identification is not completed at one time. It should be carried out regularly and programmatically throughout the project. In the project risk identification process, the most important thing is to not miss the risk factors, especially the factors that have a major impact on the overall project objectives.

The risks that exist in the development of software projects are diverse, both current and potential in the future, both internal and external, both static and dynamic, etc. Our task of risk identification is to find out the main risks that the project faces from the complicated environment.

Risk Identification Process

In order to identify risks, we first consulted relevant data and initially understood the basic concepts of risk identification. On the one hand, risk identification can be judged through perceptual knowledge and historical experience. On the other hand, it can be analyzed, summarized, and organized through records of various objective data and risk accidents. It can also conduct necessary expert visits to find out obvious and potential risks and their loss patterns. Because of the variability of risks, risk identification is a continuous and systematic task. Risk managers are required to pay close attention to changes in existing risks and discover new risks at any time.

In theory, any method that can help the risk information discovery can be used as a tool for risk identification. There are many tools and methods for risk identification. After in-depth and detailed discussion of the specific conditions of the Sunstate software project, we are at risk. The following method was used for identification:

Brain Storming

We adopted a more classic brain storming method in risk identification, through the form of a conference to give full play to the participants' creative thinking, divergent thinking and expert experience to identify project risks. The ultimate goal is to obtain a comprehensive list of risks for further clarity in the future risk analysis process. On the one hand, we try to motivate everyone's creativity in the group's decision-making process and generate as many as possible ideas. On the other hand, we question the group's ideas one by one and analyze their practical feasibility. Our group used this method at the first meeting to identify possible risks in the Sunstate software development process. Predetermine a host in advance, the host clearly clarifies the status of the Sunstate software project to all the participants and explain the meeting rules, try to create a relaxed

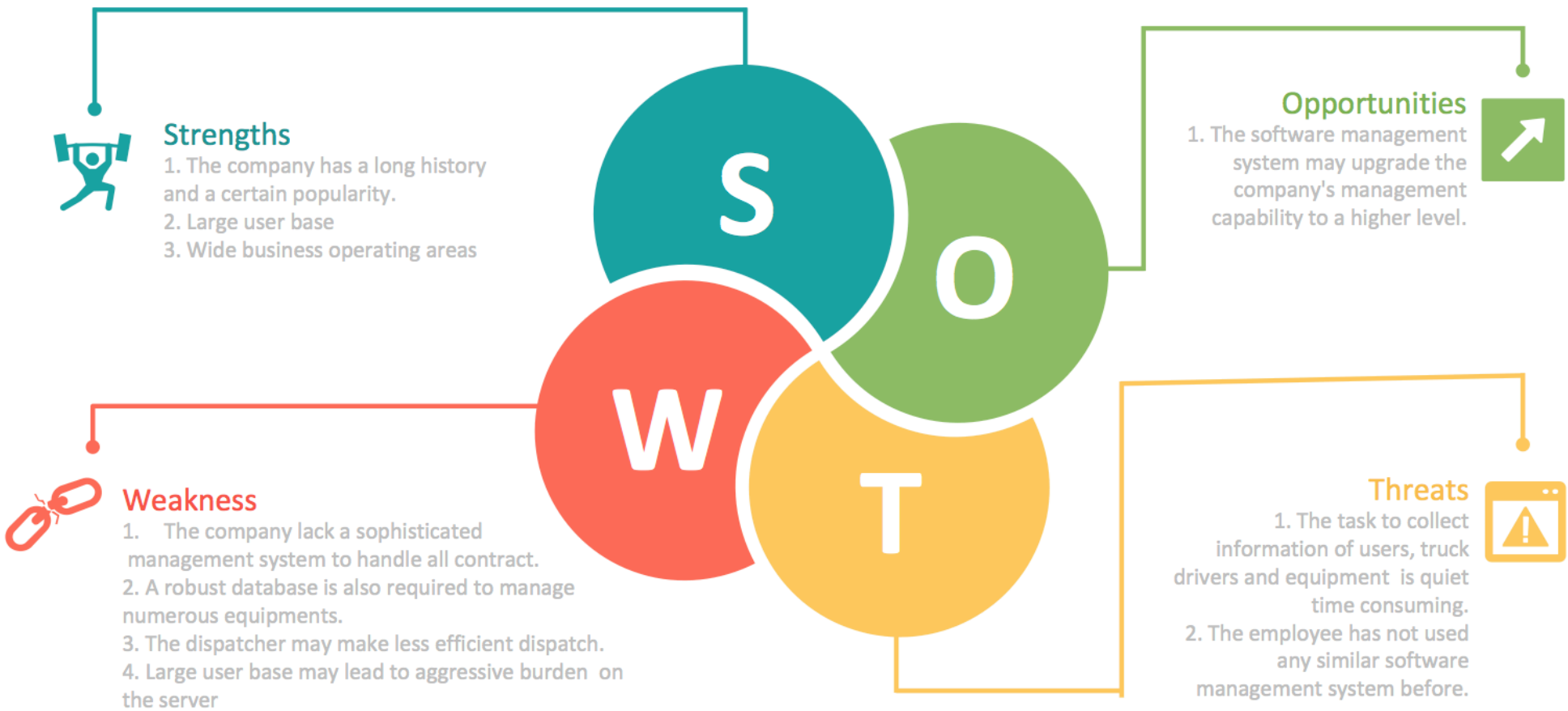
atmosphere of the meeting, encourage everyone to put forward as many risks as possible, after the end of the record to all the opinions were categorized to form the initial risk management table.

SWOT Analysis

SWOT analysis is a widely used strategic selection method and it can naturally be used to identify project risks. SWOT is an abbreviation of English, where SW refers to the Strengths and Weaknesses of the project itself and OT refers to Opportunities and Threats. When SWOT analysis is used for project risk identification, it is a comprehensive analysis of the advantages and disadvantages of the project itself and the opportunities and threats of the external environment of the project. The SWOT analysis will make a systematic evaluation of the project and eventually achieve the purpose of identifying project risks.

We use the SWOT method to carefully study the background of the Sunstate software project and find out the strengths, weaknesses, opportunities, and threats that exist. With the SWOT tool, we can quickly and effectively find hidden problems and opportunities in the project, find solutions, and clarify the future direction of development. Through a clear and organized analysis, we categorized the issues according to their priorities, clarified what are currently urgently needed to resolve, what are the things that can be delayed a little bit, what are the obstacles to the strategic objectives, and what are the tactical issues, and we will The research objects are listed, according to the arrangement of the Daws matrix, and then using system analysis to match various factors and analyze them, from which a series of strategies are derived, which are very helpful for the later decision-making and planning.

SWOT Analysis



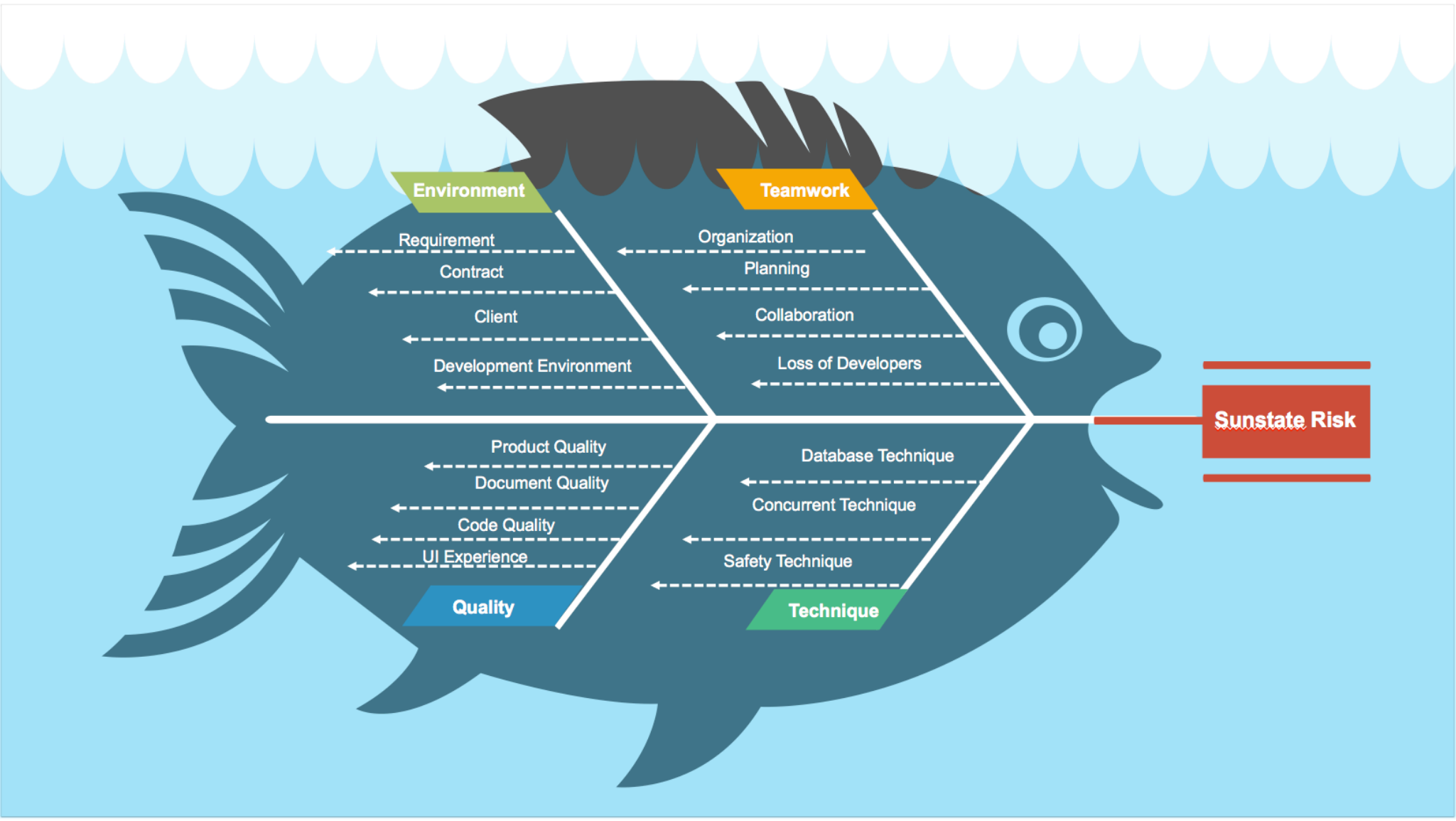
System Analysis & Fish Bone Diagram

Just based on the risks that each member thought of and summed up, we are concerned that this will still be a missed situation, so we will use the system analysis method to fully identify risks. The basic idea is to treat a complex project as a system project, decompose a complex problem into simple problems that can be easily recognized, and decompose the large system into a small system to diagnose the problem accurately.

In order to better perform system decomposition, we used the fishbone diagram tool. The characteristics of the problem are always influenced by a number of factors. We use brainstorming to find out these factors, and together with the characteristic values, the patterns that are clearly arranged, organized, and marked out according to the correlation are called the important factors. Features to figure. Because of its shape as a fish bone, it is also called a fishbone map (hereinafter referred to as a fishbone map). It is an analysis method through the phenomenon to see the essence, also called a causal analysis chart.

We use the method of drawing fishbone maps to identify risks from the perspective of the cause and analyze the factors that may threaten the project. Since the risks identified by brainstorming may overlap or cause mutual effects and may not even affect the project, the use of fishbone maps can further refine the identified risks, remove some redundant or duplicate risks, and conduct preliminary Classification. In this classification, we continue to break down the risks of the Sunstate software project into the following areas: requirements, planning, funding, personnel, development environment, technologies, processes, and products. According to each sub-risk for further analysis and discussion, this risk identification is more perfect.

Fish Bone Diagram



Risk Checklist & Delphi Method

After several rounds of work in the previous rounds, we have collated and compiled a detailed list of 30 risks under each sub-category and formed a Checklist. At the same time, in order to ensure the quality of this risk identification, we drew on the Delphi method to

conduct further evaluation and screening of these 30 risks; that is, we used a back-to-back communication method in group meetings to solicit the opinions of each member to vote (A total of 15 votes per person can be cast). Eventually, the person in charge counts the 15 highest risk votes as the result of our risk identification. During the voting, we comprehensively considered the probability and frequency of possible risks, factors such as the impact of risk on the progress, product quality, and triggers of risks. We have brainstormed to ensure the fairness and rationality of the final results.

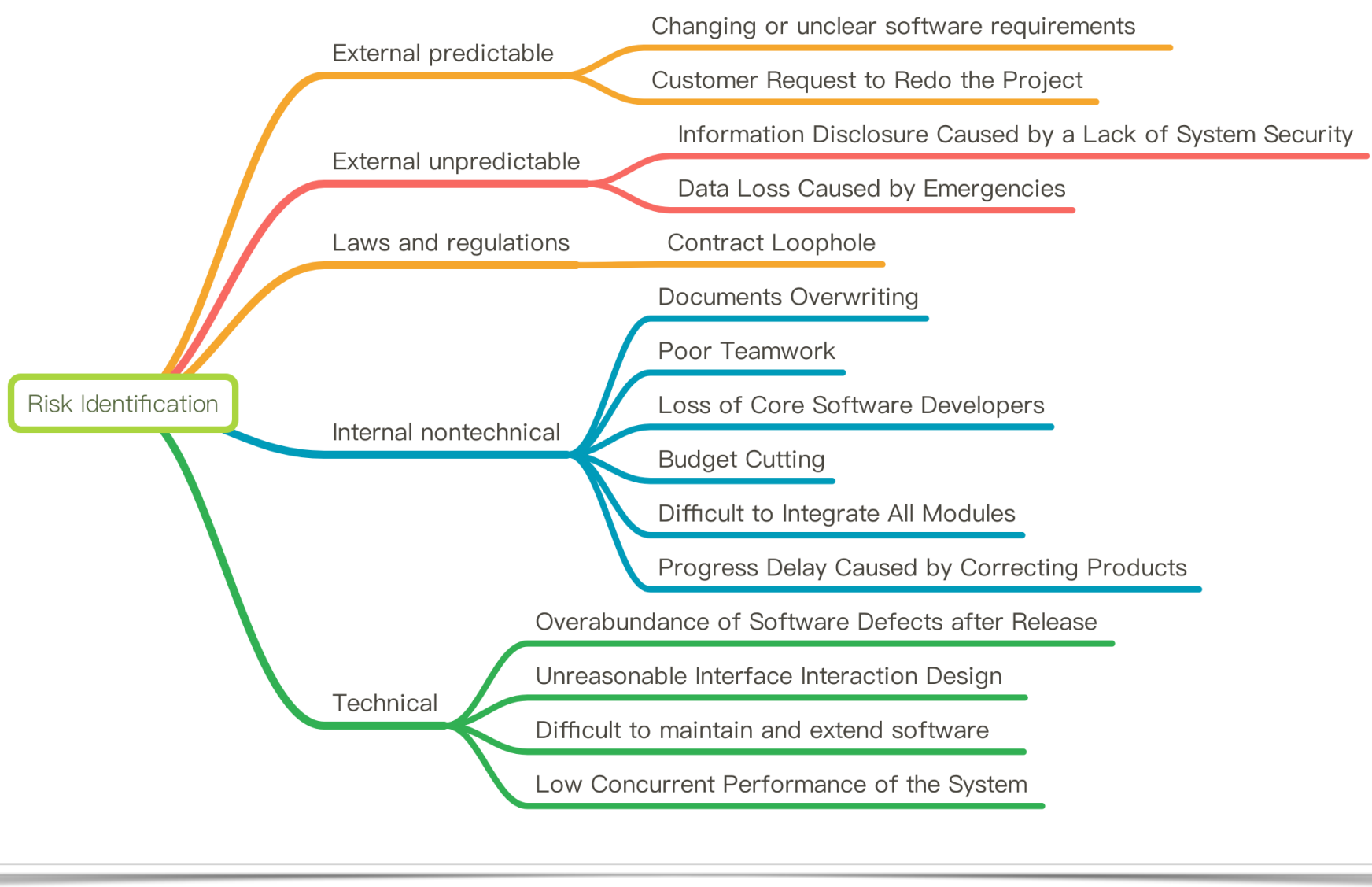
Part of Our Risk Checklist

Risk Checklist							
Risk Description	Class	X	L	W	C	S	
The requirements defined in the software requirement analysis are not clear, and further definition will expand the scope of the project; after the software development reaches a certain stage, A suddenly needs to change the demand.	Demand risk	1	1	1	1	1	5
t size (number of code lines, function points, etc.) is greater than estimated, increasing development time, delaying p	Planning risk		1				1
niliar with the product area of equipment rental software, spending more time on design and implementation than e			1	1			2
After the software development reaches a certain stage, the project budget is cut due to management decision-making or external emergencies. The original project plan is blocked due to limited funds, and major adjustments are required.	Organizing Risk	1	1	1	1	1	5
Poor communication between members of the software development team and ineffective collaboration	Staff Risk	1	1	1		1	4
ware development team lacks the necessary capabilities and quality, affecting the development schedule and produc					1		1
The loss of core personnel during the project development process led to the stagnation of the project. The company was unable to recruit new workers who could perform the core tasks in a short period of time, and the customer had a definite deadline for the completion of the project. As a result, the company's reputation was damaged.		1	1	1	1		4
velopment tools are not as effective as expected, developers need time to create work environment or switch new to	Development environment risk	1					1
andled deficiencies in the IDE used for software development and maintenance, resulting in stagnation of the develo						1	1
Separately developed modules cannot be effectively integrated and need to be redesigned or produced	Design and Implementation Risk	1		1		1	3
Designed algorithms are inefficient and affect software performance					1		1
participate in the review of the planning, prototype and specification stages, resulting in unstable demand and chang	Client Risk			1			1
me of the customer's response (such as answering or clarifying the time related to the requirement) is longer than exp		1					1

Risk Breakdown Structure

Finally, we use the risk breakdown structure to test the 15 risks from the perspective of different kinds of causes, and we also make appropriate modifications, additions and improvements according to the situation so that various types of drivers are taken into account and each risk is also taken into consideration and clear enough as well.

Risk Breakdown Structure



Risk Assessment

Acceptable Risks

Qualitative Analysis

Quantitative Analysis

List of Risks Sorted by Risk Level

Acceptable risks

Acceptable risks are risks that not worth tracking or mitigating.

Acceptable risks
Product of over size
Overtime designing due to wrong expectation
Lack of vital ability
Tools with low efficiency
Absence of customer in designing
The response from customer is time out
Extra function lengthen the process
Too mush test in compatibility
Without good structure
Low speed of loading

Qualitative analysis

Input & Output

Input

- Risk register
- Risk Management Plan
- Project scope statement
- Organization process assets

Output

- Updated risk register
  - Category
  - Priority
  - Urgency
  - Sorting
- List of risks that need to be adopted in the near future
- List of risks that need further analysis and response
- Low priority risk watch list
- Trends in qualitative analysis of risks

Methods of qualitative analysis

Delphi method

In order to analyze the level of each risk, we decided to use the Delphi method, which means that we do not discuss with each other and do not make horizontal connections between the team members. We determine that a risk manager will use the communication method to develop the Sunstate online leasing system. The risk to be assessed is sent to each member individually. Each member analyzes and evaluates each risk in the risk list by probabilistic assessment and impact assessment to obtain the risk level. The risk management personnel pass multiple rounds of survey to the members. The view of risk level was repeatedly consulted, summarized, and revised until the opinions raised by everyone became stable. Finally, they were summarized into basically the same opinions of the members. As a result of the risk level, this method has wide representativeness and the results are more reliable.

Risk probability

The risk probability must be greater than 0 and less than 1, otherwise it is not a risk. For different risks and risk situations, we use appropriate qualitative and quantitative methods to determine the risk probability. The risk probability range is mapped to a risk probability level, which is used to facilitate the risk level calculation.

Risk Probability Level

Grade	Probability range	Level description	Description
10	Over 90%	Very high	Most likely, or almost certainly
9	80%-90%		
8	70%-80%	High	Very likely to happen
7	60%-70%		
6	50%-60%	medium	May happen
5	40%-50%		
4	30%-40%	low	Less likely to happen
3	20%-30%		
2	10%-20%	Extremely low	Almost impossible
1	0%-10%		

Risk impact

The risk impact is a measure of the impact of risks. For a risk, the risk impact is also important. If the degree of risk impact is large, even if the probability of occurrence is small, it will have a very large impact on the project and cause serious consequences. Therefore, such risks must be strictly controlled. Certain risk effects are easy to measure. For example, cost risk can be measured by financial data. Progress risk can be measured by days. Some risks cannot be directly measured by numbers, such as technical risks. Therefore, we set the standard for the degree of influence and determine the risk impact of certain risks through analysis and judgment.

Risk Impact Level

Grade	Level instructions	Description
10	Veryhigh	Lead to failure
9		
8	high	Severe impact on development
7		
6	Moderate	Have more influence on project development
5		

4	low	Little impact on project development
3		
2	Vary-low	Impact on project development is negligible
1		

Level of Risk

The risk level is the most important data obtained in the risk assessment work. Based on the risk probability and the risk impact degree, the risk is classified according to the severity level through the risk level. This is an important basis for the formulation and follow-up of our risk response measures.

Calculation Method:

$$\text{Risk Level} = \text{Risk Probability} \times \text{Risk}$$

Risk Level Decision

Risk Level	Risk level	Result of decision
(Cardinal)	(Ordinal)	
81-100	Very-high	Unacceptable risks
61-80	high	Undesired risks require managerial decisions
41-60	Moderate	Risks that can be controlled need managers to review
21-40	low	Acceptable risks without review

Probability Impact Matrix

		Probability impact matrix						
	Very high	R6、R10				R9		
	High		R4、R5、R11	R1、R12	R3、R14			
	medium	R13	R2	R7				
impact	low	R8、R15					Tolerance line	
	Extremely low							
		Extremely low	low	medium	High	Very high		
				Probability				

Quantitative analysis

Input & Output

Input

- Risk register
- Risk Management Plan
- Project scope statement
- Organization process assets
- Project Management PlanOutput

Output

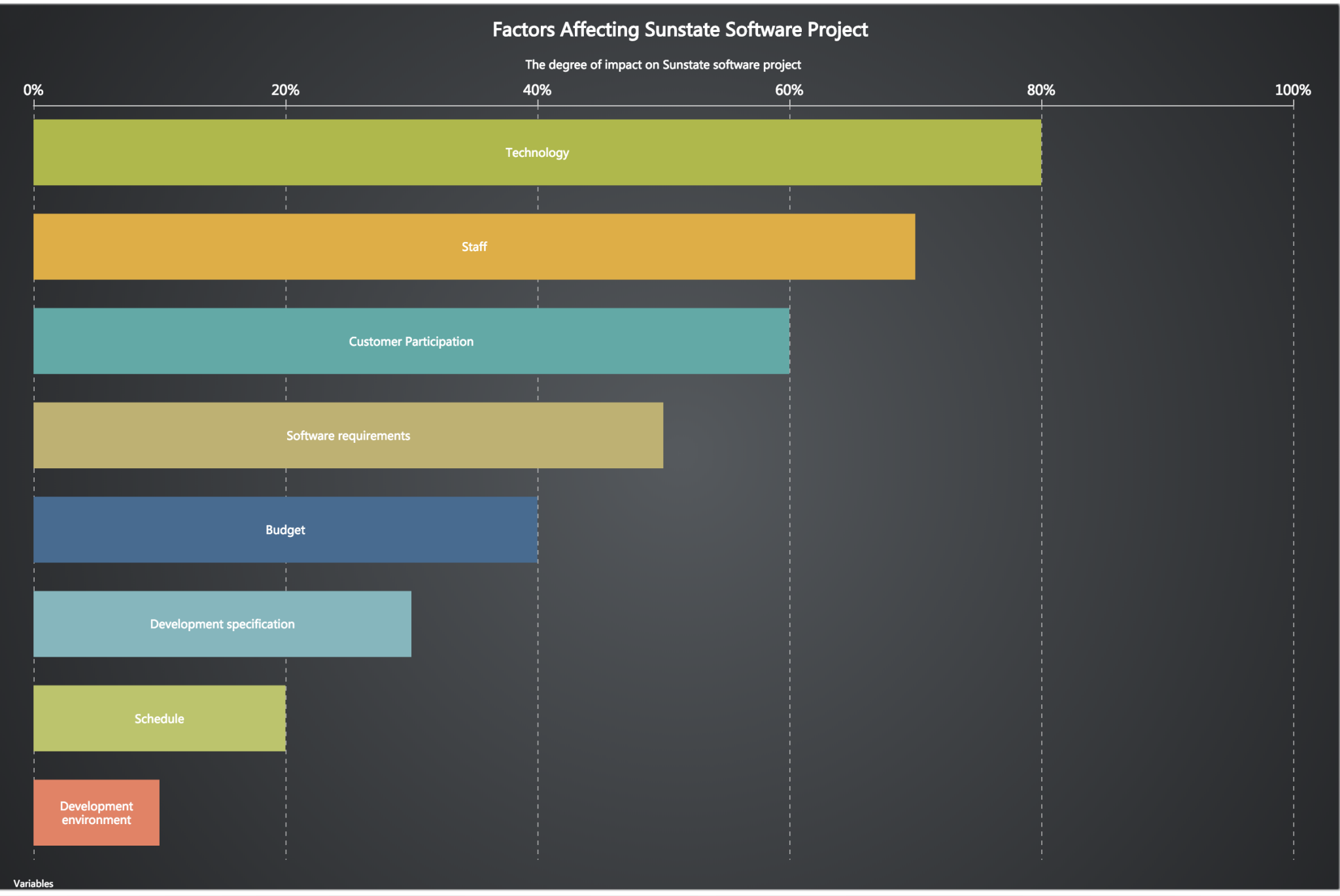
- Updated risk register
  - Category
  - Priority
  - Urgency

Methods of quantitative analysis

Sensitivity analysis

This method is to find out the sensitive factors affecting the development of Sunstate's online leasing system from a number of uncertainties, and to analyze the degree of impact on the development and the degree of sensitivity to determine the ability of the development activities to withstand risks. Uncertain analysis methods. When conducting risk assessment, one of the multiple risk factors is usually used as the input variable, and other factors are assumed to remain normal. If the small change in the factor has a greater impact on the risk outcome, it indicates that it is a sensitive factor, and vice versa, if the factor changes. If it does not affect the risk outcome or the effect is small, it is an insensitivity factor. This helps to focus on the most important factors and collect data on a priority basis. Our tool used in this Sunstate software development risk management is a tornado chart.

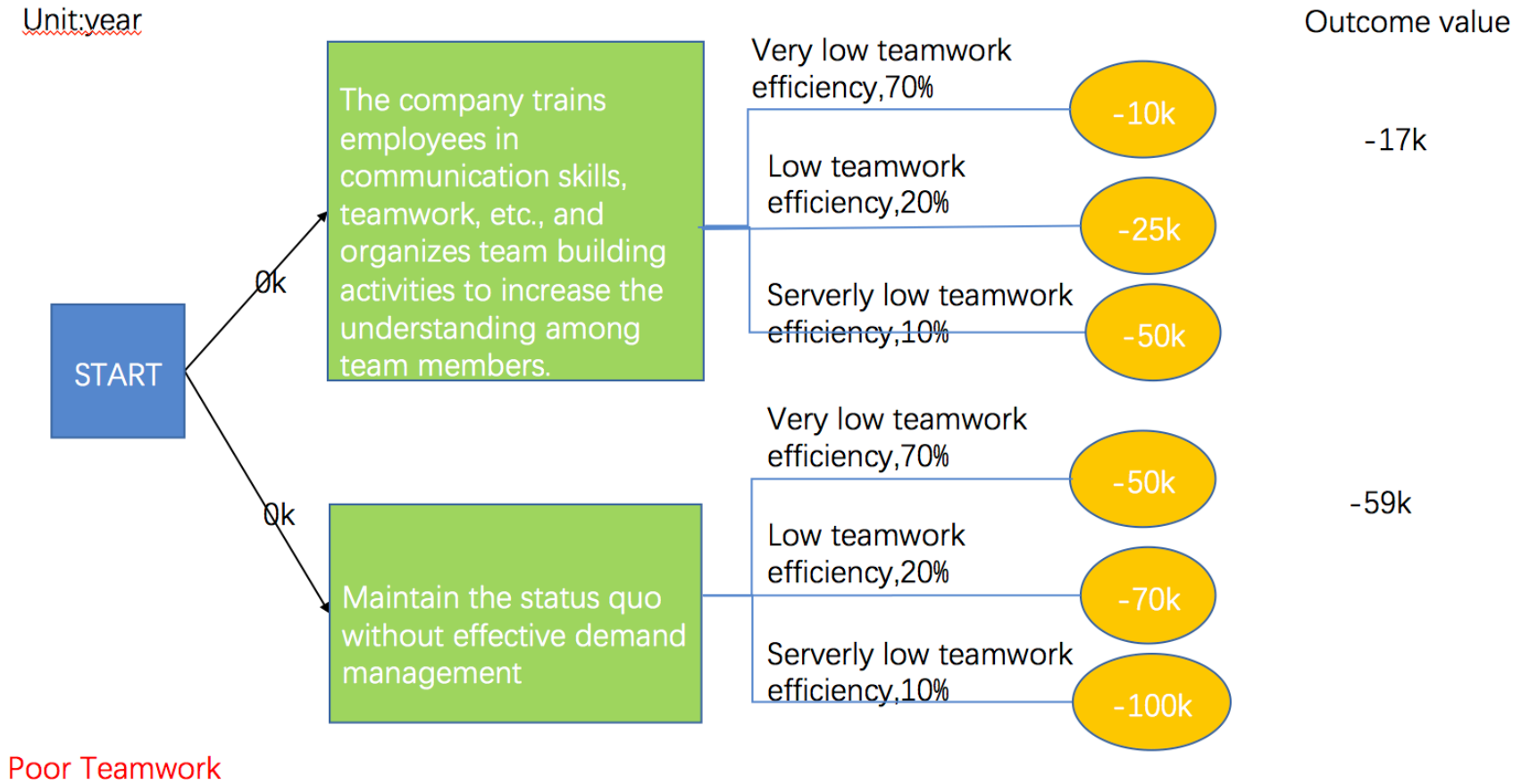
Tornado Diagram

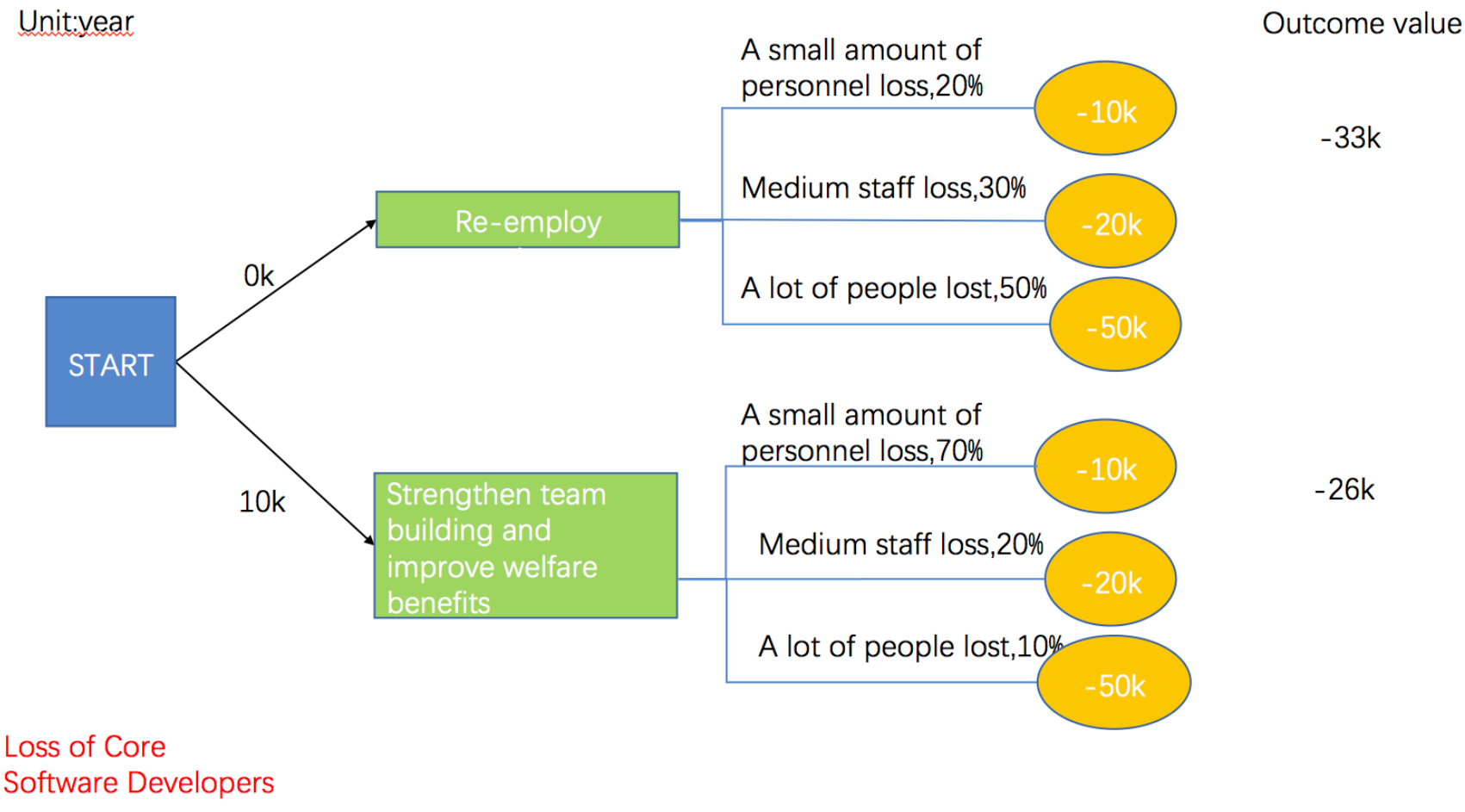


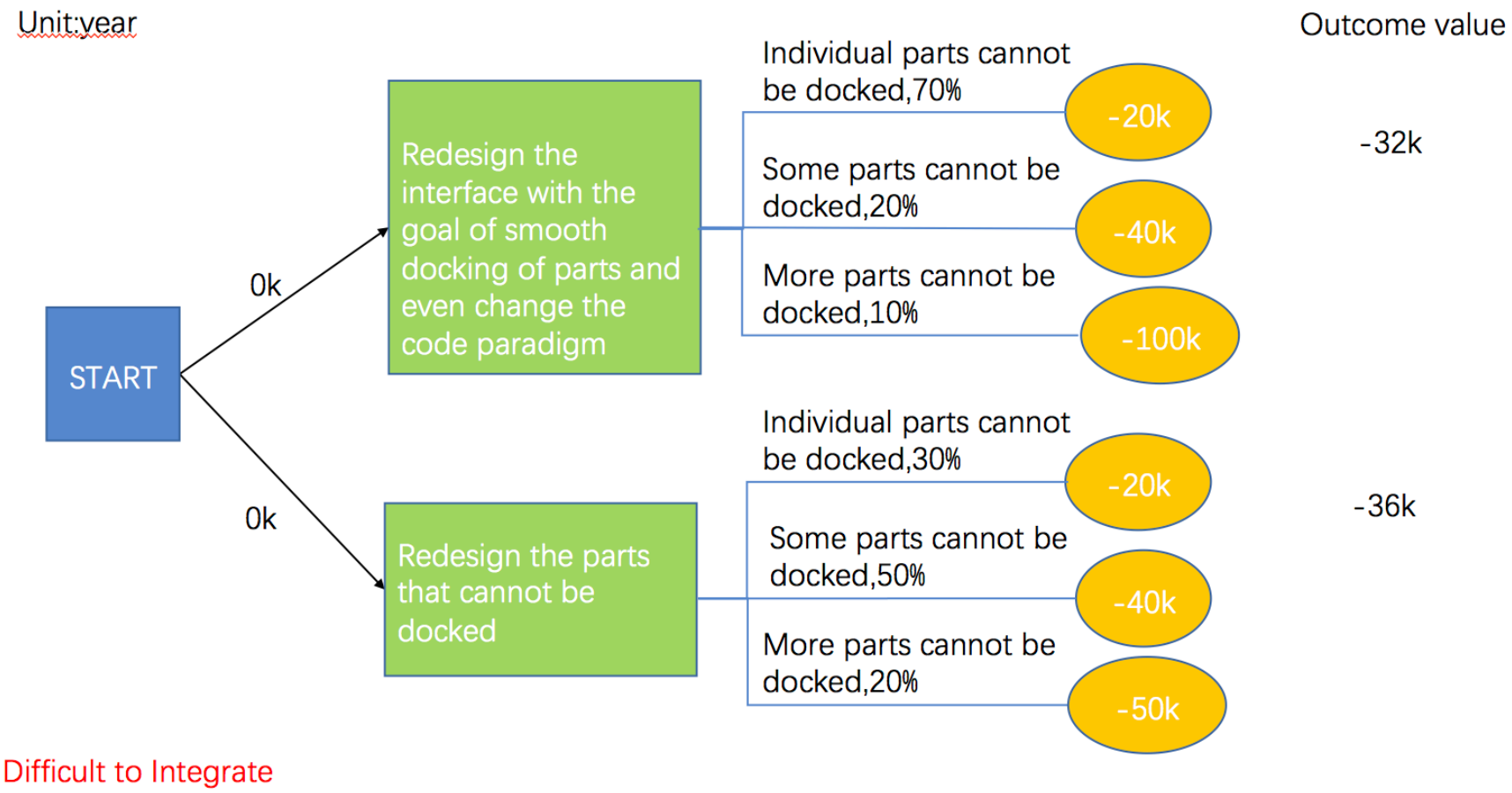
Decision tree analysis

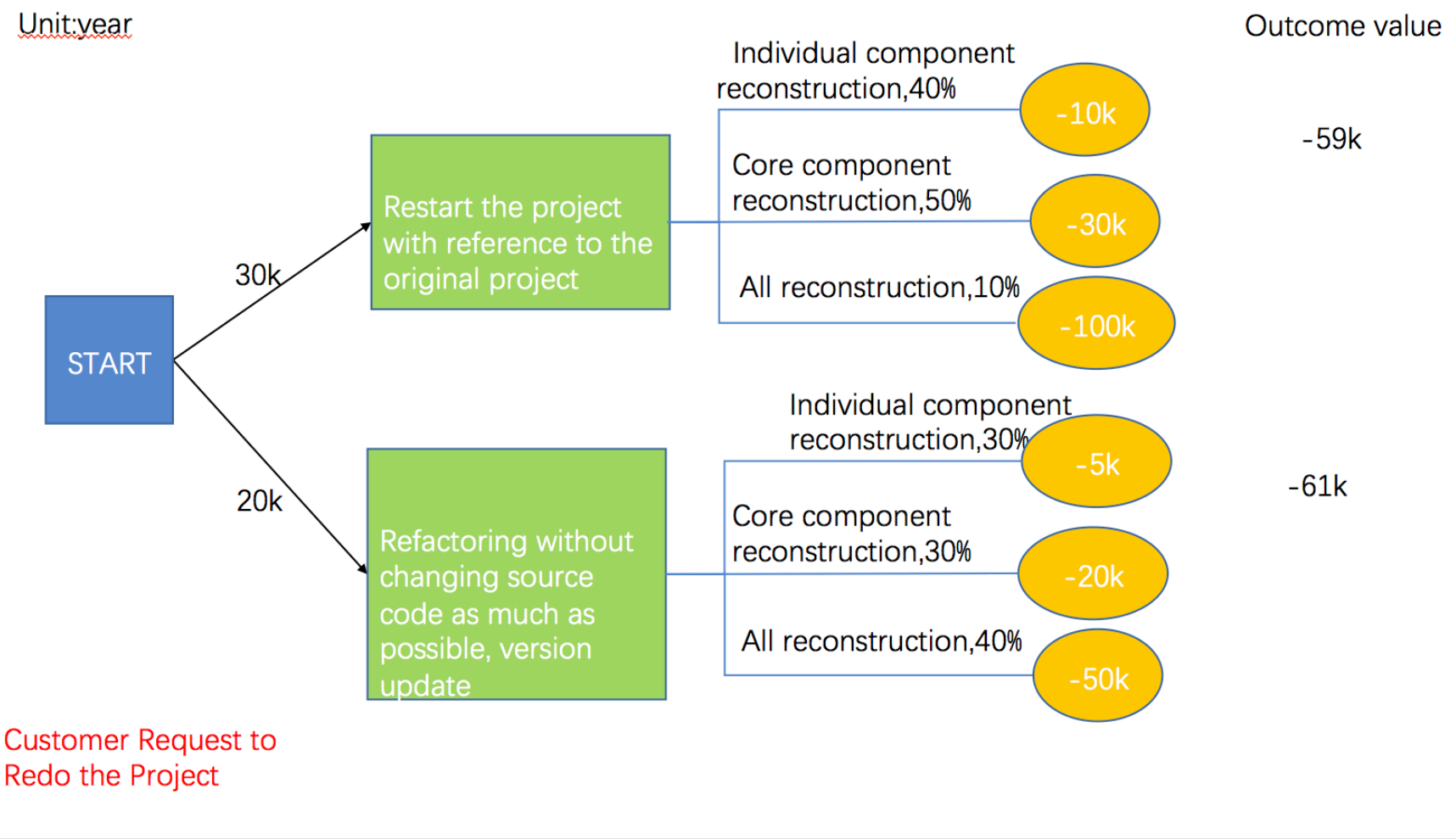
Decision tree analysis is a commonly used risk analysis decision method. This method is a method of describing the calculation, comparison, and selection of each program's future revenue in a tree-like graph. Its decision is based on the expectation value. People may encounter several different situations in the future. Each situation has the possibility of being present. People are not sure about it now, but the probability of occurrence of various natural states can be inferred from past data.

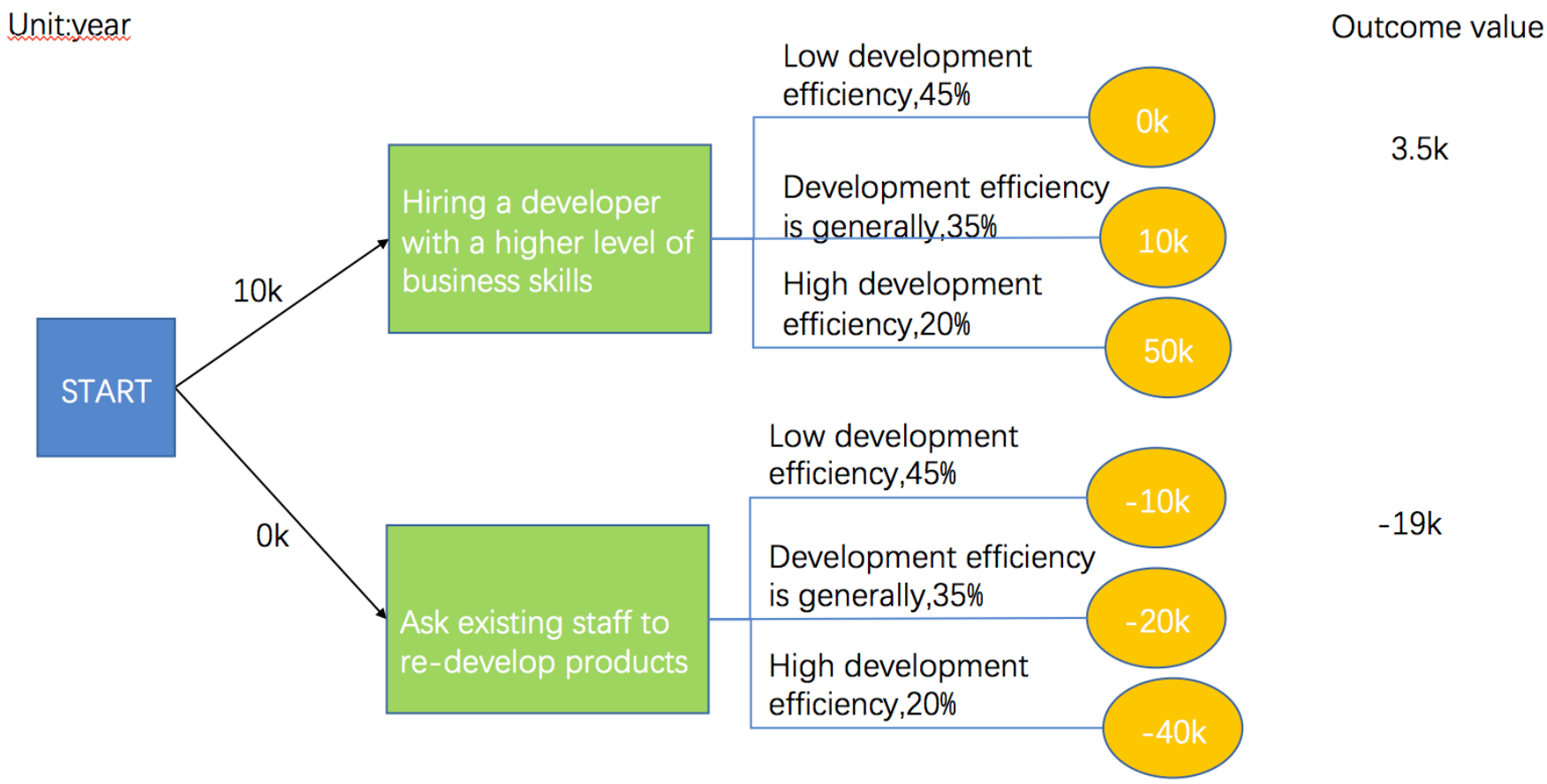
See the document: Decision Tree Analysis.


Unit:year
Outcome value
START
0k
The company trains employees in communication skills, teamwork, etc., and organizes team building activities to increase the understanding among team members.
Very low teamwork efficiency,70%
-10k
Low teamwork efficiency,20%
-25k
Serverly low teamwork efficiency,10%
-50k
-17k
0k
Maintain the status quo without effective demand management
Very low teamwork efficiency,70%
-50k
Low teamwork efficiency,20%
-70k
Serverly low teamwork efficiency,10%
-100k
-59k
Poor Teamwork

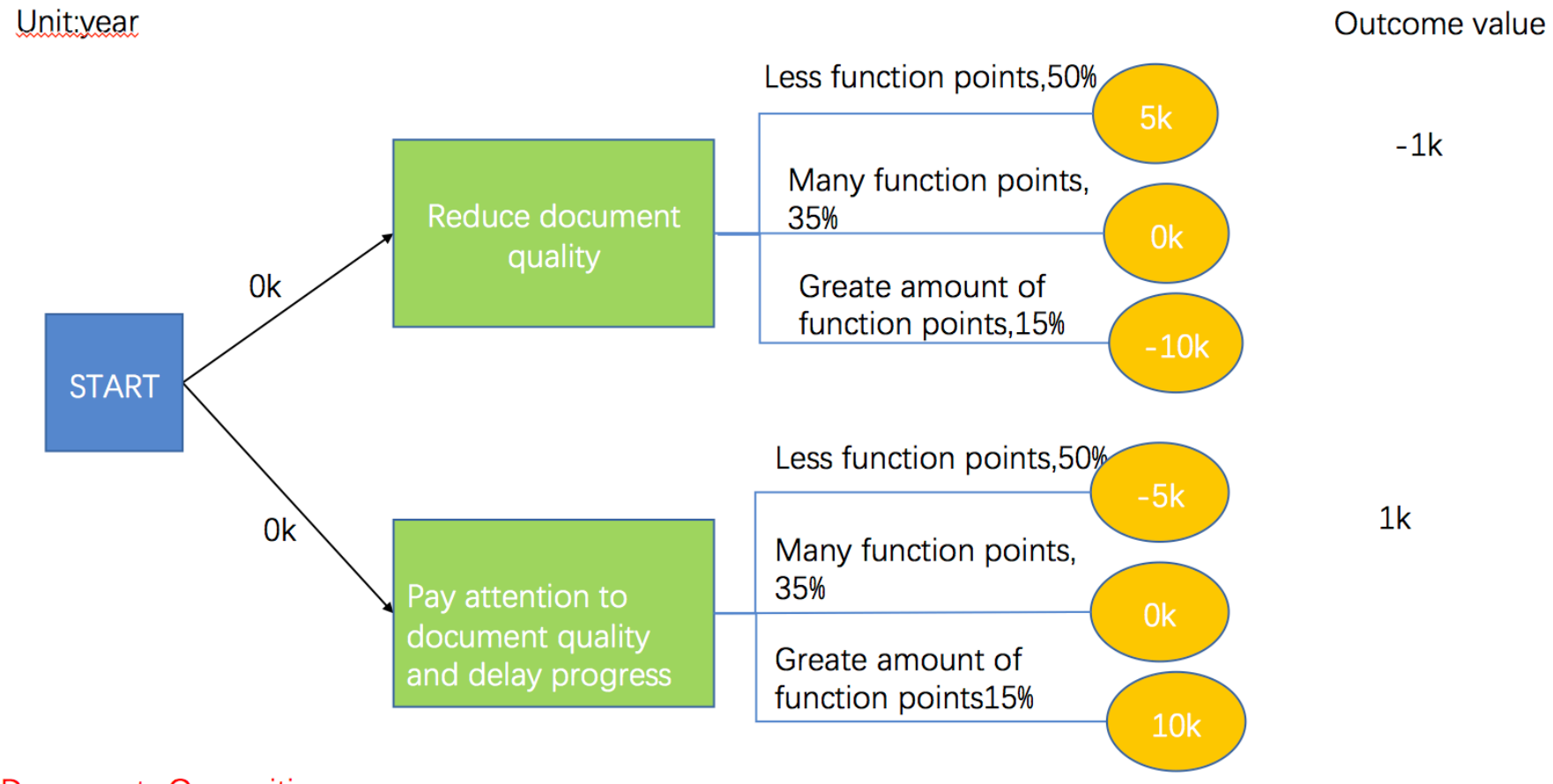

Unit:year
Outcome value
START
0k
Re-employ
A small amount of personnel loss,20%
-10k
Medium staff loss,30%
-20k
A lot of people lost,50%
-50k
-33k
10k
Strengthen team building and improve welfare benefits
A small amount of personnel loss,70%
-10k
Medium staff loss,20%
-20k
A lot of people lost,10%
-50k
-26k
Loss of Core Software Developers


Unit:year
Outcome value
START
0k
Redesign the interface with the goal of smooth docking of parts and even change the code paradigm
Individual parts cannot be docked,70%
-20k
Some parts cannot be docked,20%
-40k
More parts cannot be docked,10%
-100k
-32k
0k
Redesign the parts that cannot be docked
Individual parts cannot be docked,30%
-20k
Some parts cannot be docked,50%
-40k
More parts cannot be docked,20%
-50k
-36k
Difficult to Integrate
All Modules

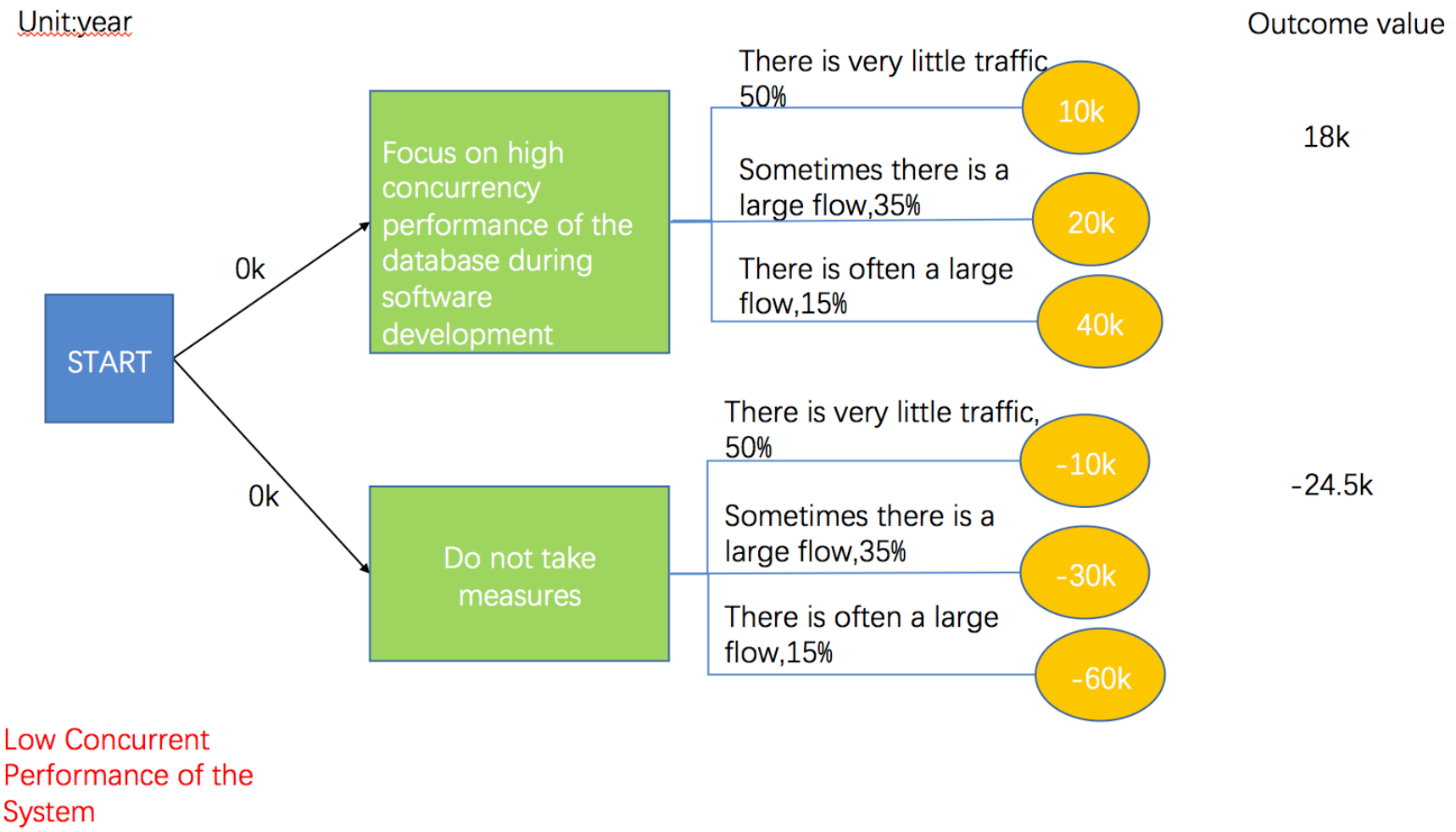

Unit:year
Outcome value
START
30k
Restart the project with reference to the original project
Individual component reconstruction,40%
-10k
Core component reconstruction,50%
-30k
All reconstruction,10%
-100k
-59k
20k
Refactoring without changing source code as much as possible, version update
Individual component reconstruction,30%
-5k
Core component reconstruction,30%
-20k
All reconstruction,40%
-50k
-61k
Customer Request to
Redo the Project

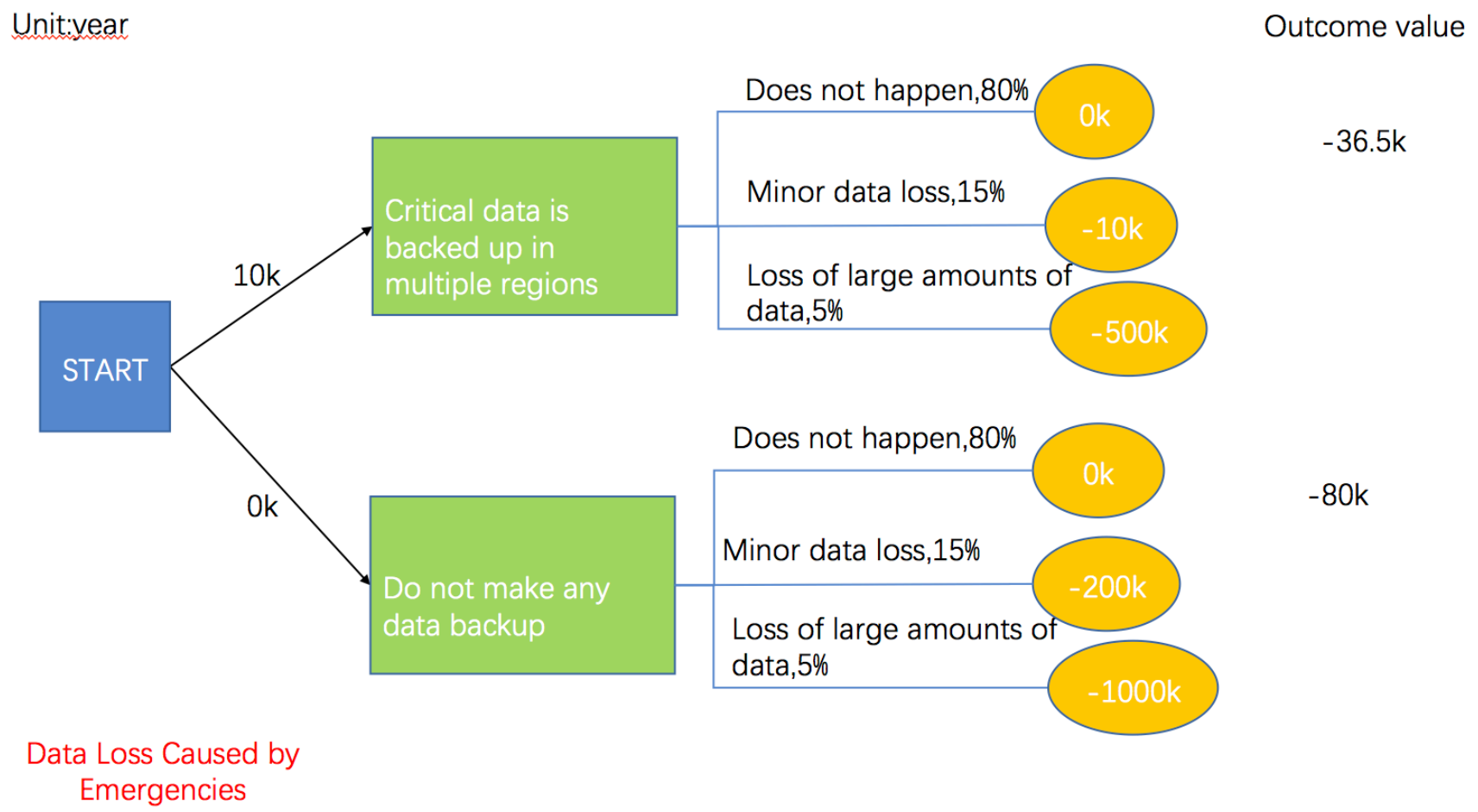


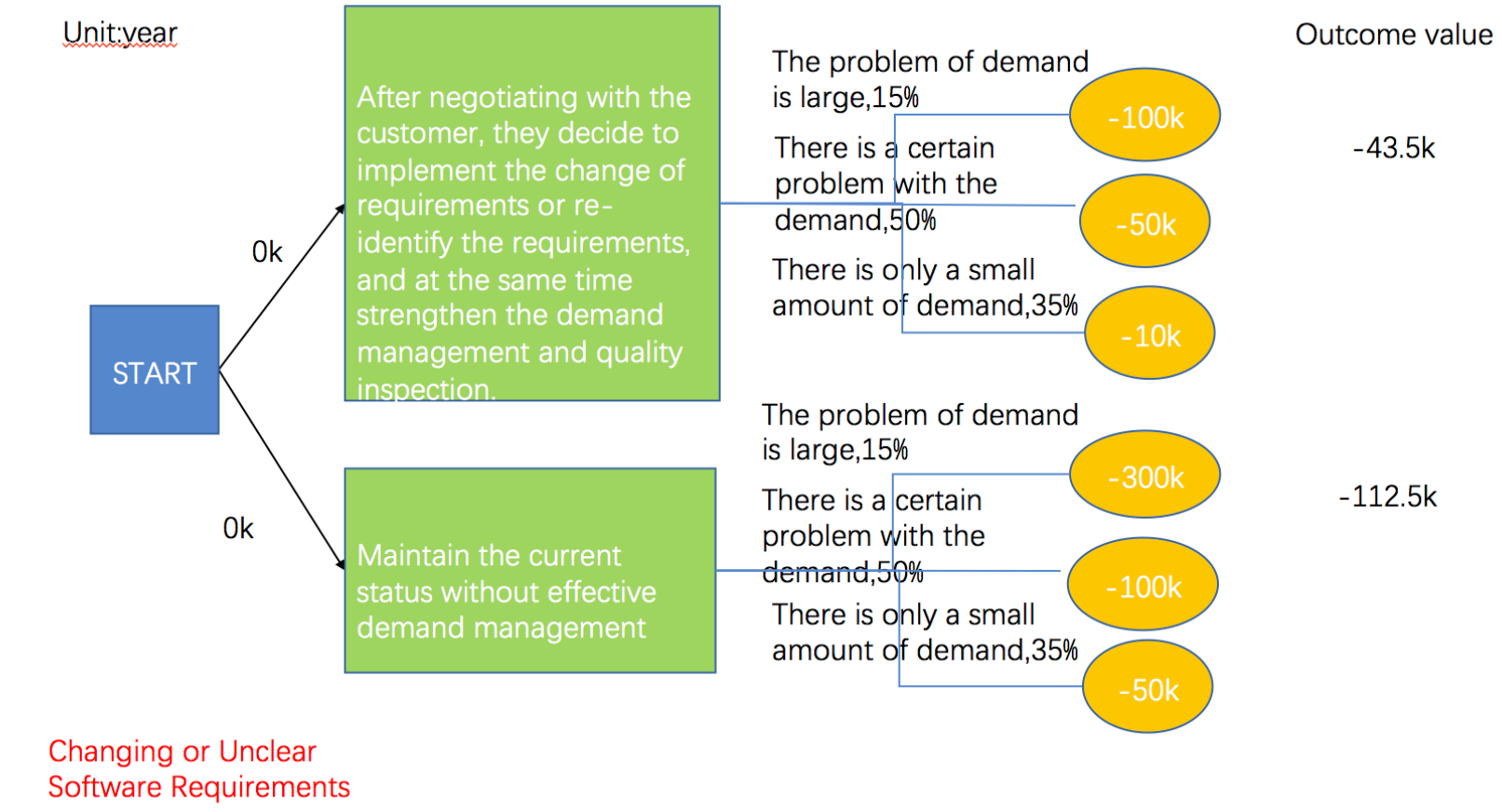
Progress Delay Caused by Correcting Products

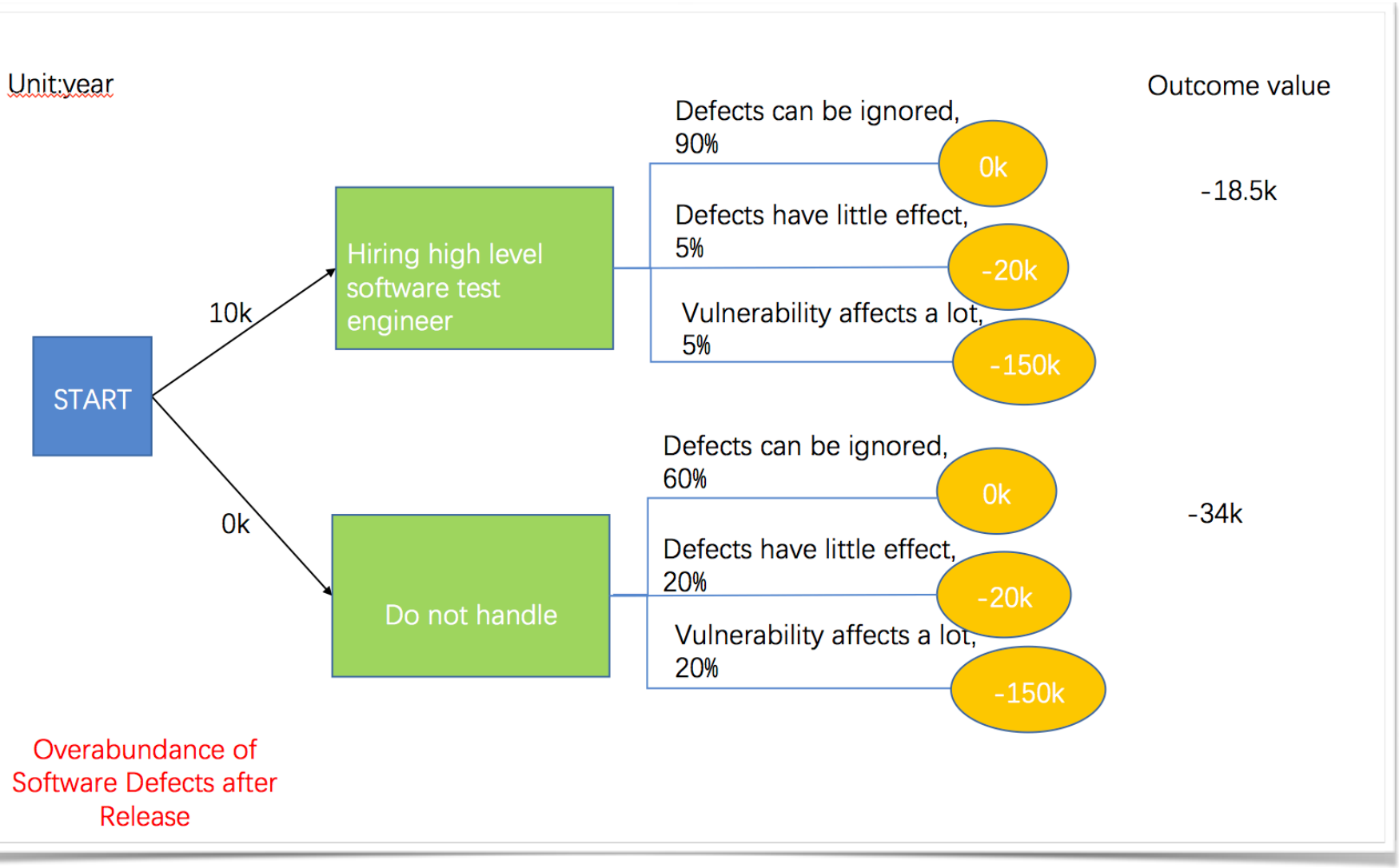


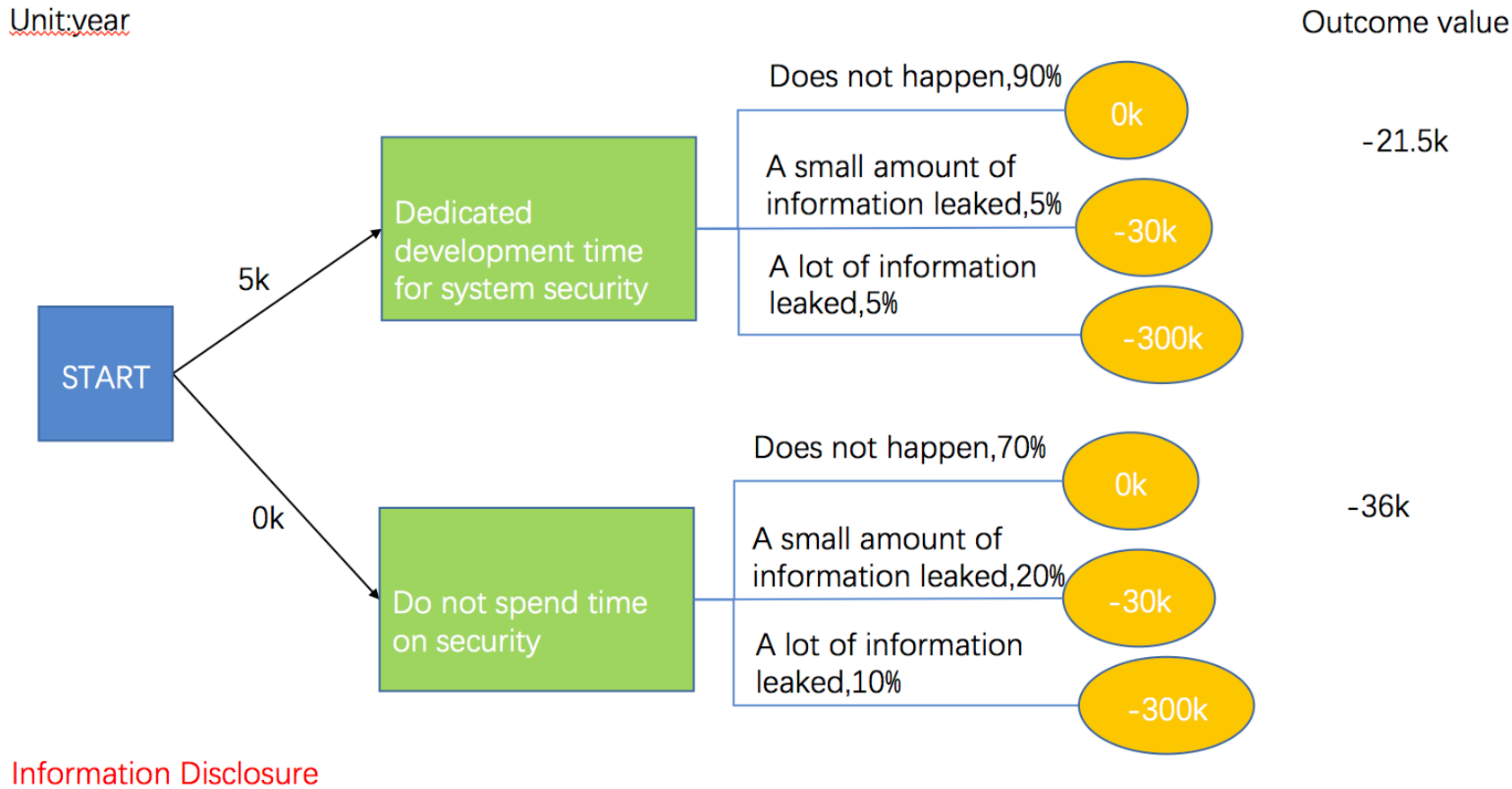
Documents Overwriting

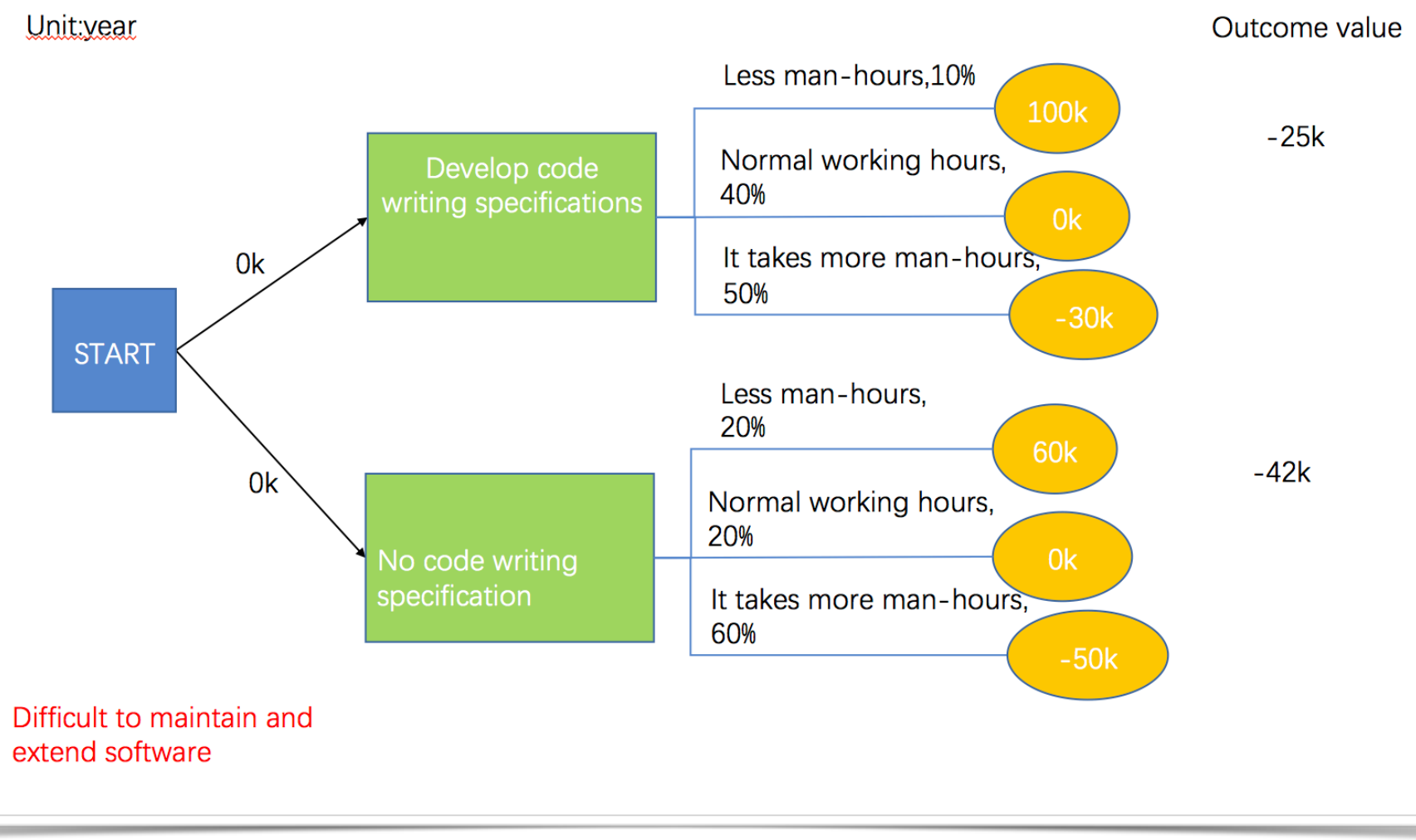

Unit:year
Outcome value
START
0k
Focus on high concurrency performance of the database during software development
There is very little traffic 50%
10k
Sometimes there is a large flow,35%
20k
There is often a large flow,15%
40k
18k
0k
Do not take measures
There is very little traffic, 50%
-10k
Sometimes there is a large flow,35%
-30k
There is often a large flow,15%
-60k
-24.5k
Low Concurrent Performance of the System

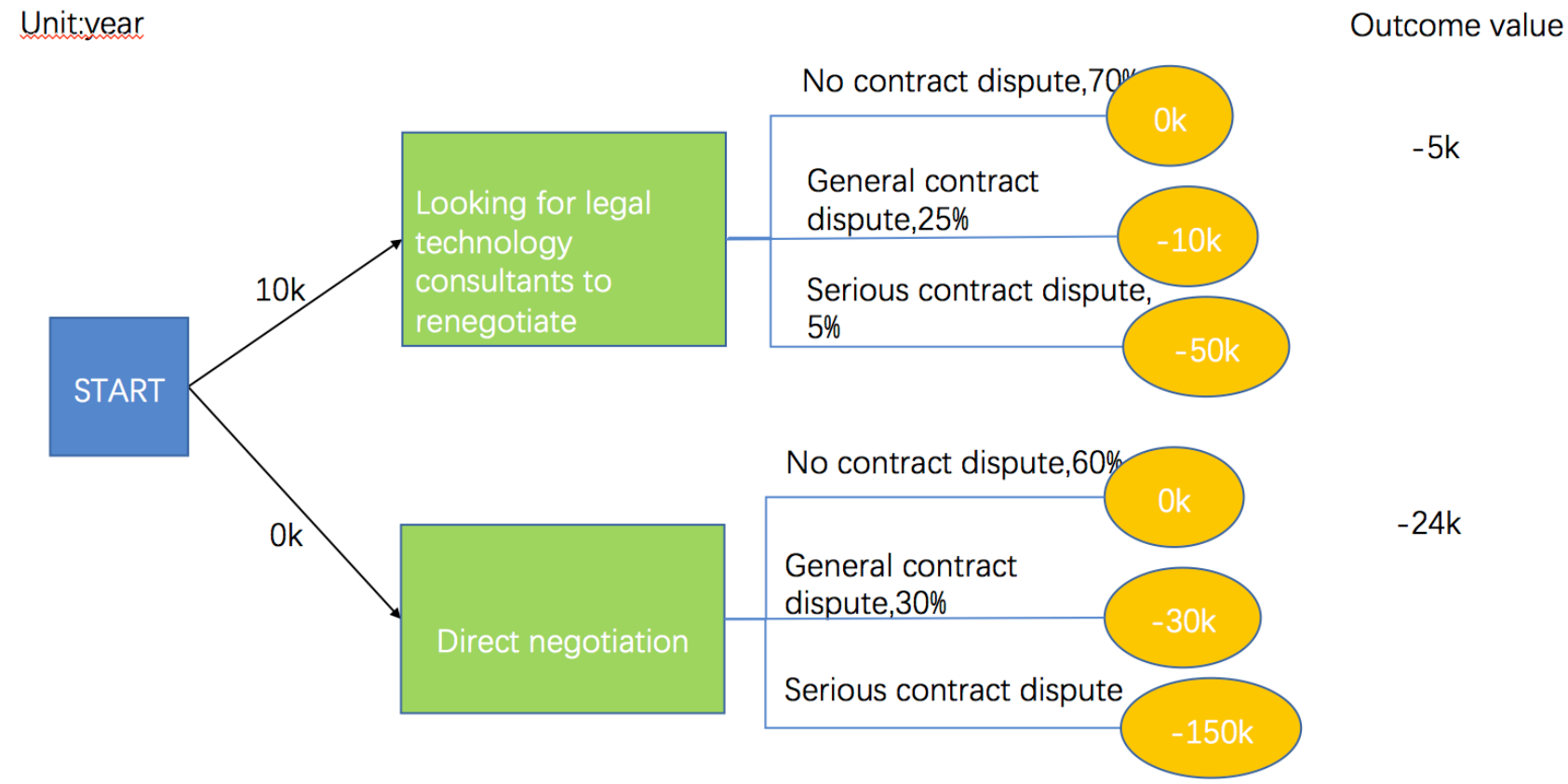

Unit:year
Outcome value
START
10k
Critical data is backed up in multiple regions
Does not happen,80%
0k
Minor data loss,15%
-10k
Loss of large amounts of data,5%
-500k
-36.5k
0k
Do not make any data backup
Does not happen,80%
0k
Minor data loss,15%
-200k
Loss of large amounts of data,5%
-1000k
-80k
Data Loss Caused by Emergencies


Unit:year
Outcome value
START
0k
After negotiating with the customer, they decide to implement the change of requirements or re-identify the requirements, and at the same time strengthen the demand management and quality inspection.
The problem of demand is large,15%
-100k
There is a certain problem with the demand,50%
-50k
There is only a small amount of demand,35%
-10k
-43.5k
0k
Maintain the current status without effective demand management
The problem of demand is large,15%
-300k
There is a certain problem with the demand,50%
-100k
There is only a small amount of demand,35%
-50k
-112.5k
Changing or Unclear Software Requirements

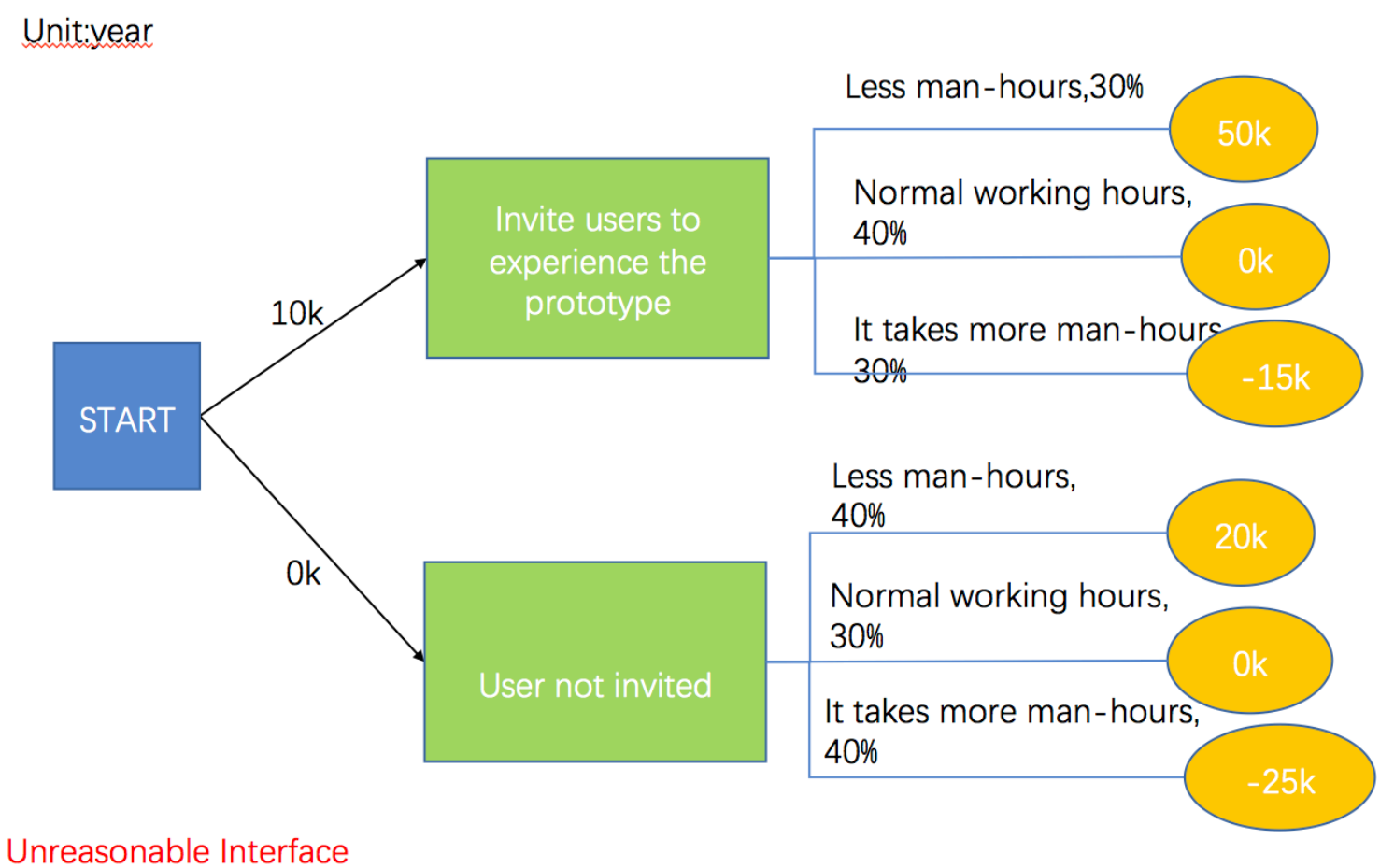

Unit:year
Outcome value
START
10k
Hiring high level software test engineer
Defects can be ignored, 90%
0k
Defects have little effect, 5%
-20k
Vulnerability affects a lot, 5%
-150k
-18.5k
0k
Do not handle
Defects can be ignored, 60%
0k
Defects have little effect, 20%
-20k
Vulnerability affects a lot, 20%
-150k
-34k
Overabundance of Software Defects after Release


Unit:year
Outcome value
START
5k
0k
Dedicated development time for system security
Do not spend time on security
Does not happen,90%
0k
A small amount of information leaked,5%
-30k
A lot of information leaked,5%
-300k
-21.5k
Does not happen,70%
0k
A small amount of information leaked,20%
-30k
A lot of information leaked,10%
-300k
-36k
Information Disclosure Caused by a Lack of System Security

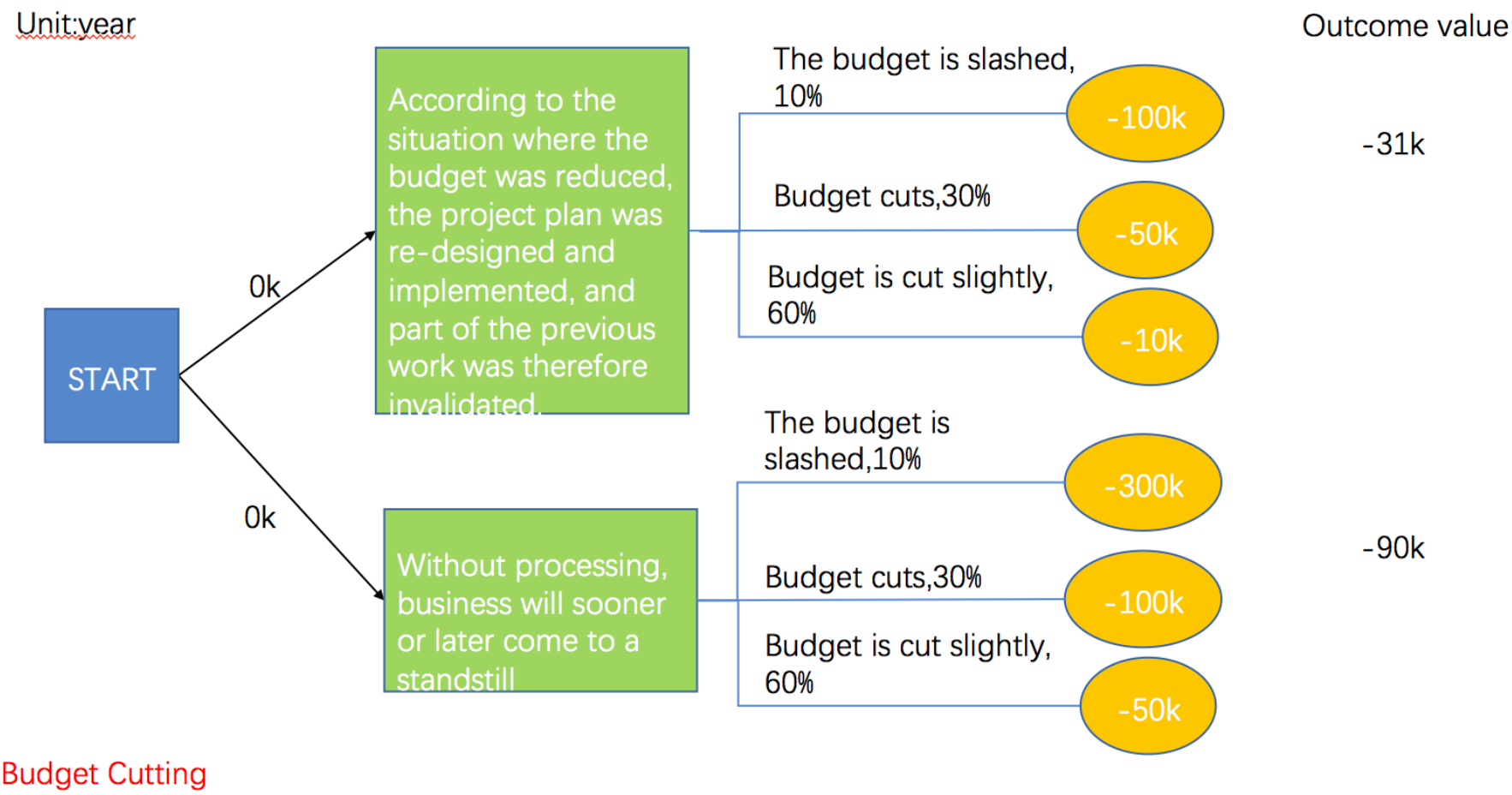

Unit:year
Outcome value
START
0k
0k
Develop code writing specifications
No code writing specification
Less man-hours,10%
100k
Normal working hours, 40%
0k
It takes more man-hours, 50%
-30k
-25k
Less man-hours, 20%
60k
Normal working hours, 20%
0k
It takes more man-hours, 60%
-50k
-42k
Difficult to maintain and extend software



Contract Loopholes



Unreasonable Interface Interaction Design



Budget Cutting

List of risks sorted by risk level

No.	Risk event	Risk category	Risk level	Outcome value
R9	Low system concurrency	Technical risk	81	18k
R14	Software maintenance and expansion difficulty	Software maintenance risk	56	-25k
R3	Ineffective teamwork	Staff risk	49	-17k
R1	Changing or unclear software requirements	Requirement risk	48	-43.5k
R12	Overabundance of software defects after release	Test risk	35	-18.5k
R11	Inadequate system security causes information disclosure	Technical risk	32	-21.5k
R5	Modules cannot be integrated	Design and Implementation Risk	28	-32k
R7	Correcting Products Delays Progress	Product risk	25	3.5k
R4	Loss of core developers	Staff risk	21	-44k
R6	Customer request to redo the project	Customer risk	20	-59k
R2	Budget cuts disrupt original project plan	Organization management risk	18	-31k
R13	Contract loophole	Contract risk	12	-5k
R10	Loss of database data	Technical risk	10	-36.5k
R15	Unreasonable interface interaction design	Software use risk	8	0.5k
R8	Overwriting documents	Process risk	3	1k

For more detail, please refer to the document: Risk Register

Risk Response Plan

Input & Output

Low Concurrent Performance of the System

Overabundance of Defects after Software Release

Poor Teamwork

Difficult to Maintain and Extend Software

Changing or Unclear Software Requirements

Input & Output

Input

- Risk register
- Risk Management Plan

Output

- Risk Management Plan —— Person liable
- Project Management Plan Update
- Contract update (risk related contract agreement)
- Organization Process Asset Updates

Risk Response Plan

Low Concurrent Performance of the System

Why?

If a company has a solid customer base, it means that the management system has a high probability of dealing with large volume orders. If the server side of the system can't effectively buffer and handle such large traffic, the client's request processing will be slow, and the worst case will make the whole system paralysed. The concurrent performance of Sunstate's equipment leasing software will affect the processing effect of the system on frequent order requests.

How?

The solution is sought through data and examples of related distributed cache database and high concurrent applications.

What?

A project has just started to implement the basic functions. With the iteration of the version and function, large data and high concurrency are the problems that the software design must consider.

The following are the four directions for solving high concurrency.

The path is short

- Staticize pages- Users can get pages directly, without having to go through so many processes.
- Use caching - extract data from the database for the first time, then save it in the cache, and then extract data directly from the cache. However, mechanisms are needed to maintain consistency between cache and database.
- Use Stored Procedure - Those processes that require multiple visits to the database for one request can integrate operations into the Stored Procedure, so that only one database access can be done.
- Batch read - when multiple concurrent requests are combined to reduce the number of databases accessed.
- Deferred update - In the case of high concurrency, multiple requests can be kept in the cache first, and then the data in the cache can be kept in the database at a time. The risk is that the data in the cache may be lost.
- Use index - Indexes can be regarded as special caching. Using index as much as possible requires that the value of index columns accurately be given in the where clause.

The data to be querying is few

- Sub table - the content of the same table can be divided into multiple tables according to the area and category. It is a very simple idea, but try to avoid the multiple table query .
- Separation of active data - for example, there are a lot of registered users for logon services, but active logon users are very few, so we can save a list of active users. When quering, we query active table first, and then check the master table if not, which is similar to caching.
- Database partition – We optimize in the database layer, which is transparent to the program, so querying large data only needs to find the corresponding block.

Shunting

- Cluster - when the system is facing a large number of user access, the number of servers is usually increased when the load is too high, and the processing capability of the whole system is improved by using cluster and load balancing, distributing concurrent requests to different servers, one of which can be a business server or a database server.
  - At present, many IT companies have developed a load balancing solution for themselves, for example, Maglev is Google's solution for its own data center, and is used in the production environment at 2008. At the thirteenth network system design and implementation USENIX Symposium (NSDI '16), engineers from Google, University of California at Los Angeles, and SpaceX shared the detailed information about the commercial server load equalizer Maglev. Maglev will be able to handle 1 million requests per second without preheating for 5 seconds.In Google's performance benchmarks, the Maglev instance runs on a 8 kernel CPU with a network throughput upper limit of 12M PPS (packets per second), and if Maglev uses a Linux kernel network stack, the speed will be less than 4M PPS.
  - Coincidentally, UCloud, a domestic cloud service provider, has further iterated the load balancing product, Vortex, which has successfully improved the performance of single computers. In terms of technology implementation, UCloud Vortex is quite similar to Google Maglev. As an example of an ordinary cost-effective x86 1U server, Vortex can achieve a throughput of 14M PPS (10G, 64 byte line speed), a new connection above 200K CPS, and the number of concurrent connections to 30 million and 10G line speed.
- Distributed - Assign multiple business logic of single request to multiple servers, so that many logic can be processed synchronously, which is generally used in specially complex business requests.
- CDN – Shunt in the layer of domain resolution, such as the distribution of user requests in the Southern China area to the server in Southern China, and user requests in Central China allocated to central China servers.

Lateral expansion

Using MySQL master - slave configuration to achieve separation of read and write, reduce database pressure.

Who?	Developers should be familiar with the development environment, and have certain project experience for multithreading development and caching database. DBM can manipulate related databases skillfully.
When?	Developers need to spend enough time on software development to ensure that the system can handle large traffic properly. DBM needs to daily conduct maintenance and management of database in later system operation. If the database is abnormal or unable to handle short time traffic, developers must respond quickly.
How much?	The main capital expenditure comes from the maintenance of database and the salary of developers.

Overabundance of Defects after Software Release

Why?	In the software testing phase, testing can never detect all defects, and it is impossible to test the quality of software. Many IT experts also strongly advocate that the purpose of software testing is to find software errors, and to find as many defects as possible as early as possible in the life cycle of the software development, but if the design use case is not perfect, there will be a situation where the defects of the software can not be found especially when use cases are not updated in time. Besides, if we do not use the testing technology to test every module, we will also miss many obvious defects. If there is not enough software testing to find software defects in time before software deployment, there will be a large number of problems in the use of software, and the team needs to improve the software code.
How?	Through the analysis of the existing Internet Co's coping strategies, we believe that in the requirement analysis stage, we should be careful and prudent to prevent the conflict between the functions and design the use cases by the equivalent class method and the boundary value method. With the deepening of software development or the update of software, the test cases are constantly updated, and a variety of testing tools and methods are used to conduct software testing.

What?

- **Stress test**

When the strength of the performance of Sunstate's software has not been tested, it is easy to make it impossible to meet the customer's requirement to make an appointment, which may lead to the low efficiency of the system and even the risk of the downtime of the database, so it is necessary to test the pressure. The most common APP measurement usually means that the operation team of the product uses a robotic simulation to test the APP server pressure before a network APP is available, that is, a script robot is used to simulate a user's possible operations in the program. It usually arranges thousands of robots to perform various operations in the server continuously, recording the capacity of the server and the related performance indicators in the case of high pressure. In addition, the existing pressure testing tools can be used to test the performance of the logical function of the database. In the case of Jmeter, Apache JMeter is a Java based pressure testing tool developed by the Apache organization. It is used for stress testing of software. It was originally designed for Web application testing, but later extended to other test areas. It can be used to test static and dynamic resources, such as static files, Java small service programs, CGI scripts, Java objects, databases, FTP servers, and so on. JMeter can be used to simulate huge loads on servers, networks or objects, test their strength and analyze overall performance from different pressure categories.

- **Compatibility test**

When the compatibility of App on various types is not fully tested, after the software is used, it may appear that the users of some types of models can not use app normally, and the development of Sunstate company's business is affected, so the software compatibility test must be carried out.

Currently, generic compatibility tests generally cover Top100-300 models. The deployment of compatibility testing is an empirical process that can draw on the mature App testing methods of large Internet Co. At present, Google, Facebook, BAT and other companies usually use automated testing platform to build their own test lab, and complete so many models testing.

Security test

The main purpose of security testing is to find potential security problems in software programming and check the application's ability to prevent illegal intrusion. According to the different security indicators, different testing strategies are different. If we follow the same principle, to prove the security of the software, it will be beneficial to the work of software security testing, which is beneficial to the development of software safety testing.

The security test will check the system's ability to prevent illegal intrusion. During the safety test, the tester disguised himself as an illegal intruder and tried various ways to break through the defense line.

For example,

- try to intercept or decipher a password;
- use a client software that can disintegrate any defense to attack the system;
- cause a system error with a purpose to invade the system in the process of system recovery;
- try to find the key to the system by browsing the non confidential data
- destroy the system so that others can not visit.

Theoretically, as long as there is enough time and resources, there is no system that can not be entered. Therefore, the criterion of system safety design is that the cost of illegal intrusion exceeds the value of protected information. At this time the unlawful invaders have been unprofitable.

Security testing is used to verify whether the protection mechanism integrated in the system can protect the system from illegal intrusion in practice. The security of the system must be able to withstand the frontal attack - but at the same time it must be able to withstand the side and rear attacks.

Who?	Software tester should be familiar with various testing methods and tools.
When?	The design of test cases should be started when the requirements of the project are determined, and the methods of software testing are constantly used in the development process of the project until they are delivered to the customers.
How much?	It is mainly about the time consumption of software testing for development and the salary of software tester.

Poor Teamwork

Why?	The development of Sunstate's large software engineering project requires every member of the team to perform their duties, communicate effectively and cooperate with each other. If problems arise in teamwork, it will greatly hurt the enthusiasm of internal members, and bring about low efficiency of software development or failure to guarantee software quality.
How?	By consulting the relevant literature on how the software development team collaborate efficiently, the solution is sought with the successful experience of other development teams and a variety of positive and negative cases.

What?

Poor teamwork can be shown in many aspects, such as irrational composition of the team, unreasonable or undefined division of responsibilities, low communication efficiency, and ineffective cooperation among members. Here are some risk coping strategies:

- **We need to choose people with strong communication and technical skills to form a team and arrange team structure rationally.**

The key elements of an efficient software development team are team structure and "human". Selecting team structure and personnel is a very important starting point, and these two aspects complement each other. In the selection, the general attention is paid to the communication and technical ability of the personnel, and the learning ability is also very important, but it is absolutely not the aspect that the efficient software development team needs to pay attention to. After selecting good people, we will decide what kind of structure it is based on the existing personnel. If the communication ability of the team is very strong, then the horizontal oriented architecture is the first choice; if the communication ability of most people is weak and the communication ability of the few people is strong, the vertical structure is the first choice.

- **Use rewards and punishments to improve the sense of responsibility of our members.**

For example, we can conclude a team project reward mechanism, assigning project dividends to the team members, making a float on the dividend distribution of the project when the project contract is entered into. In this way, we can fully mobilize everyone's enthusiasm in the process of development.

- **Rationalize the division of labor and make clear the responsibility.**

Teams are made up of individuals. Individuals in a team often experience different backgrounds, different backgrounds, different personalities, and different levels. Before the formation of the team and the formal start up, we should first carry out a reasonable division of labor and make full use of everyone's characteristics and hobbies to give full play to everyone's

expertise. Because if the work is unpleasant and disagreeable, the efficiency will be low. After the completion of the division of labor, the corresponding responsibilities of each person are determined. At this time, each team member should be clearly stated that it is best to assign the tasks to everyone in the form of text and conduct their daily performance assessment in order to avoid the occurrence of mutual prevarication and mutual waiting.

- **An effective communication and collaboration mechanism can be developed with agile development strategy.**

When the division of labor is completed, the team begins to work. At this time, it is necessary to ensure the unimpeded information in the whole team, especially the colleagues who have a working relationship between each other. In case of the problem, it needs to be put forward in time so as not to cause unnecessary waste of time. But software development itself is a work that needs to be concentrated and quiet, and many temporary interruptions cause a stagnation of development ideas, so team leaders should be able to organize people to communicate and understand the progress of work every day in a fixed period of time. Fixed time will help you form habits and improve efficiency.

Through frequent communication, daily standing meeting, feedback and other ways to solve the problem of poor communication and lack of cooperation; through leadership, programming in pair, code collective ownership and other ways to promote team cooperation, improve skill quality, cope with personnel changes, reduce contradictions and conflicts; improve people's sense of achievement and morale through frequent product launches; through on-site development, reduce the risk of lack of customer involvement.

- **Conduct regular communication skills and teamwork training, organize group building activities to enhance tacit understanding between members.**

- **Use Trello, Github and other tools for effective management of collaborative development.**

- **Check regularly and adjust it in time.**

The bottleneck of the team may change, so the team leader needs to assess the situation and adjust in time. For example, modify the previous division of labor, and even replace the incompetent team members.

Who?

Team members should have good communication skills, technical ability, and have a strong sense of collectives.

Project managers must have certain project management experience, understand the level and progress of personnel among different parts, and be able to adjust the best collaborative plan at any time.

When?

Members should be selected for the project team while selecting a reasonable team structure.

After the team is established, the rewards and punishment mechanism and communication cooperation mechanism should be clearly defined as soon as possible, and an efficient collaborative development management tool should be selected.

In the process of software development, small team training and construction should be held regularly; while the team atmosphere and work progress should be closely monitored, and timely adjustment for more efficient collaboration should be made.

How much?

The main cost of funding comes from team members and software costs, such as hiring experienced software project managers, organizing training and regiments, and paying for efficient collaborative development management tools.

Difficult to Maintain and Extend Software

Why?	Considering risk from the point of view of the project maintenance, if the quality of the project code is not in good quality, the low extensibility and poor readable code will make it difficult to maintain the project when there is a need to increase the function of the project in the later period. And if the maintainer is not the original developer, people need to have a new understanding of the architecture of the project, the implementation of the code before maintain the project. Poor code readability can seriously affect the efficiency of maintenance, especially in the face of fierce market competition, so we need to maintain and update the software of Sunstate company regularly.
How?	Through consulting the relevant literature to improve the code quality, we make corresponding management for code writing.

What? Implementation of code quality monitoring

- **Code Review Conference**

The specific implementation measures are as follows: a weekly review conference is conducted at the initial stage of the project. Developers are randomly selected to state the business modules developed this week.

- **Check the code daily**

Understand the progress of the developers in the morning meeting. The technical manager opens the code and checks and records the problems. Feedback to the team leader in time. At the beginning of the development phase, one copy of the code is taken out of the document every three days of the work day. In the later development, the code is reduced to five working days according to the degree of knowledge that the developer is familiar with in the code.

- **Statistical analysis of code defects**

At the beginning of the testing phase, the code defects are analyzed statistically. The contents of the statistical analysis are as follows:

- bug module statistics.
- bug statistical statistical items: whether design is unreasonable, whether coding is not standard, testing environment.
- bug level statistical analysis: the severity level of bug.

Use of code static testing tools

Use code static detection tools to control code quality. The common tools are: checkstyle, PMD, FindBugs, etc.

Improve the project document

Write better project documents, from project structure to specific implementation, so that maintenance personnel can understand projects more easily.

	Set up a special SQA group The SQA team performs specific quality assurance and quality control tasks (including technical review and multilevel testing strategy) to control the quality of all software processes.
Who?	Programmers need to be strict with themselves and have some coding experience to ensure the quality of code.
When?	Code quality should be emphasized in the code preparation phase to prevent future problems.
How much?	The main cost of money is to pay the salaries of excellent programmers; The writing of good code and annotations slows down the project coding process. It also takes a long time to make documents for the necessary functions.

Changing or Unclear Software Requirements

Why?

Customers connect to the development team and give the development team a certain reward to open the project, but most customers do not know the development, maintenance, and testing mechanism of a project.

Customers can provide only demands, and these needs are often not closely linked to actual development, so the making requirements specification becomes a very critical process for customers to communicate with the development team. As for the leasing software developed for Sunstate, business needs need to be focused on the business logic of leasing.

Requirements change has an important impact on software development. We can neither refuse customers' change requests nor accommodate customers blindly, so we must control well before we need to change the requirements. The purpose of demand change control is not to control the occurrence of change, but to manage changes and ensure orderly change.

How?

- **Change prevention at the start of the project**

Ensure that the requirements are basically clear and there will be little change in the future.

- **Change of demands in the implementation phase of the project**

In the early stage of the project, a set of corresponding measures should be predefined in order to take corresponding measures according to the degree of change of demand, and corresponding measures should be taken in accordance with the degree of change when the software demands change really happens.

What?

- **Clear the contract constraints and establish a demand baseline**

The impact of change in demand is obvious to the software development, so when signing a contract with a customer, a number of relevant terms can be added, such as defining the time for a customer to change the demand, setting up what kind of change can be accepted, rejected or partially accepted, and it is also possible to enforce change management when the demand is changed.

Although it is difficult for software development contracts to define each requirement accurately at the beginning of the contract, the binding force of the contract can not be ignored. Clarified and established the baseline of demand is the basis of demand change. In the process of development, the first requirement baseline is established after the needs are identified and evaluated. After each change and review, the new requirement baseline should be redefined, so that small demand can be changed, but the general direction should be protected from frequent changes. For example, for project needs, hierarchical management can be implemented to achieve control and management of demand changes.

- **Establish a change examination and approval process**

In practice, people often do not want to carry out formal demand management processes for small demand changes, and think it reduces development efficiency and wastes time. It is this idea that makes demand change uncontrollable and ultimately leads to project failure. Therefore, small demand changes have to go through the formal demand management process, otherwise they will accumulate a lot to be problems.

We should define the approval process, examination and approval personnel, approval matters. There are two purposes to do this: first, standardize the process of changing requirements as far as possible, and reduce unnecessary, non-urgent, and unreasonable invalid changes. The second point is to leave a written basis for future possible cost changes and claims. Any change that fails to fulfill the examination and approval procedures will be invalid and should not be accepted.

- **Manage changes in hierarchical management and with regular batch processing**

In the software development project, "the customer is always right" and "the customer is God" is not completely correct, because in the signed project contract, any new demand changes and increases not only influences the normal progress of the project, but also affects the cost of the customer's cost and income. Therefore, it is not a good thing for users to constantly propose requirements that have a significant impact on project progress. When customers meet demands, we can not refuse to develop them blindly. Therefore, when the customer insists on changing the new demand, it can be suggested that the customer divide the new demand according to the level of importance and urgency as a basis for the evaluation of demand change. For example, a weekly or even every two week or even a monthly demand change meeting is held to focus on dealing with these fragmentary changes, take the initiative to control the pace of work, and try to avoid affecting the progress of the project as a result of handling fragmentary changes. For the outcome of the meeting, a demand change plan can be formally submitted to the customer, indicating the time, cost, time limit and increase of workload caused by the change. It is requested that the customer should match the demand change plan, determine the time limit for change, control the scale of alteration.

- **Arrange full-time staff to be responsible for change management**

Sometimes the development task is heavier, developers are easy to fall into the development work and ignore the communication with customers at any time. Therefore, we need to arrange a full-time demand change liaison officer to be responsible for timely communication with customers and track and report the progress and situation of the demand change. At the same time, the project change control group can be set up to decide which changes are accepted. The group is composed of the multiple personnel involved in the project, which should include the clients and the decision-makers of the developers.

- **Confirm that the customer accept the cost of change**

To let customers realize that changes are costly, we need to work with our customers to determine whether the change of requirements is still in progress. For example, when there is no problem with the change, we should make it clear whether the customer can accept such problems as delay in schedule, increase in cost and efficiency. Generally speaking, if the customer considers that the change is necessary (not the leader of his superior), he will accept these consequences. By consulting with customers, the development team will not incur complaints from the company and customers even if there is no return. If the customer considers the change necessary but can be postponed, the two sides will sign the memorandum until later. If the customer considers the change to be dispensable, the change will be cancelled in most cases. This prevents frequent changes and allows customers to realize that not all requirements need to be changed.

Who?	Customers need to express their needs clearly at the beginning, and actively communicate with developers to explore the possibility of the completion of requirements changes. Developers should try to prevent frequent demand change by communicating with customers and taking necessary change management.
When?	At the beginning of the project, customers should discuss their demands with the software developer team. When the demand change occurred, we should carry out change management.
How much?	The main expense lies in the formulation and evaluation of the initial demand specification, and the time and capital loss brought by the adjustment of the demand when the demand changes really happen.

Risk Monitoring and Control

Input & Output

Risk Monitoring and Control Outline

Risk Monitoring and Control Process

Input & Output

Input

- Risk register
- Project Management Plan
- Approved change requests
- Job performance information
- Performance report

Output

- Requested changes
- Risk register
- Recommended corrective measures
- Recommended preventive measures
- Project Management Plan Update
- Organization Process Assets Updates

Risk Monitoring and Control Outline

Risk monitoring is a variety of supervision and control activities in the whole project, which are carried out according to the project risk management plan and the project's actual risk and project development. This is a project risk management based on the stage, gradualism and controllability of project risk, because only when people recognize the process and possibility of project risk development, the project risk is controlled. Furthermore, when people understand the main characteristics of project risks and their consequences, they can monitor project risks. It is uncontrollable when people have no knowledge of project risk.

Project risks are developing and changing. This development and change will also change with people's control behavior. The process of controlling the project risks is a process of transforming the subjective initiative to the objective world. At this time, all kinds of information produced will further improve people's understanding and mastering of the project risks, so that the people's control behavior of the project risk is more consistent with the objective law. In fact, people's monitoring of project risks is a process of constantly recognizing project risks and constantly revising project risk monitoring decisions and behaviors. This process is a process of gradually transforming project risks from controllable to controllable through people's behavior.

Risk Monitoring and Control Process

We will realize project risk monitoring by monitoring the development of project risks, identifying the symptoms of project risks, taking various risk prevention measures, dealing with and handling the occurring risk events, eliminating or narrowing the consequences of the project risk events and implementing project risk management plans.

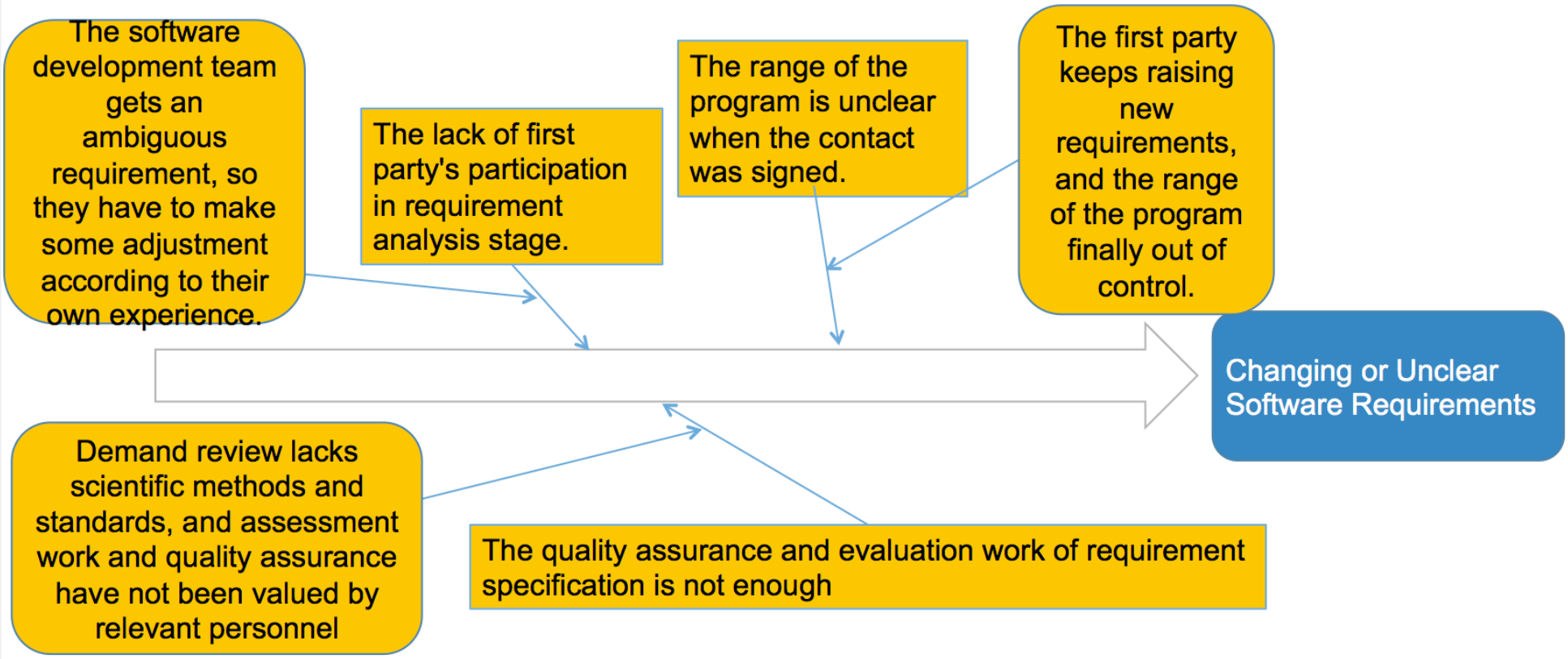
Causal Analysis Chart

In this process, we use causality analysis diagram as a monitoring tool to classify and decompose the main factors that have an impact on the risk of a certain type, so as to trace the root cause of the risk and identify the risk of the project as soon as possible. The basic principle of a causal analysis diagram is that if a project has a risk, it will take risks again unless it takes timely measures. By learning from past lessons, we can play a preventive role. The causal analysis diagram method can generally be completed by the following three stages.

- determine the cause of the risk.
- determine the measures to prevent the risk of the project.
- implement management behavior.

Through the causal analysis, we give emergency measures to deal with each risk that just happened so as to reduce the probability of risk re-occurrence, avoid the occurrence of other project risk events, and reduce the loss caused by the occurrence of the risk, and actively eliminate the negative consequences of the project risk events.

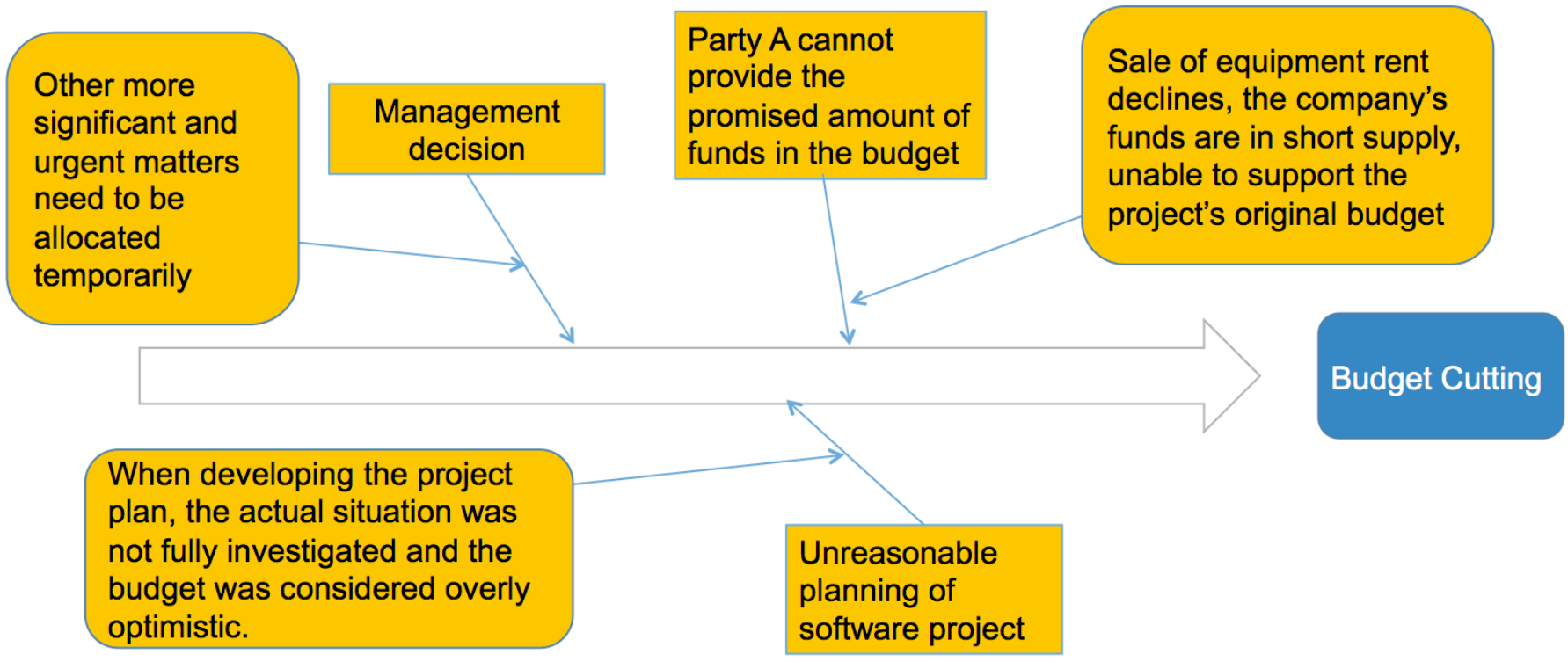
Causal Analysis Charts & Corresponding Emergency Measures



Solutions :

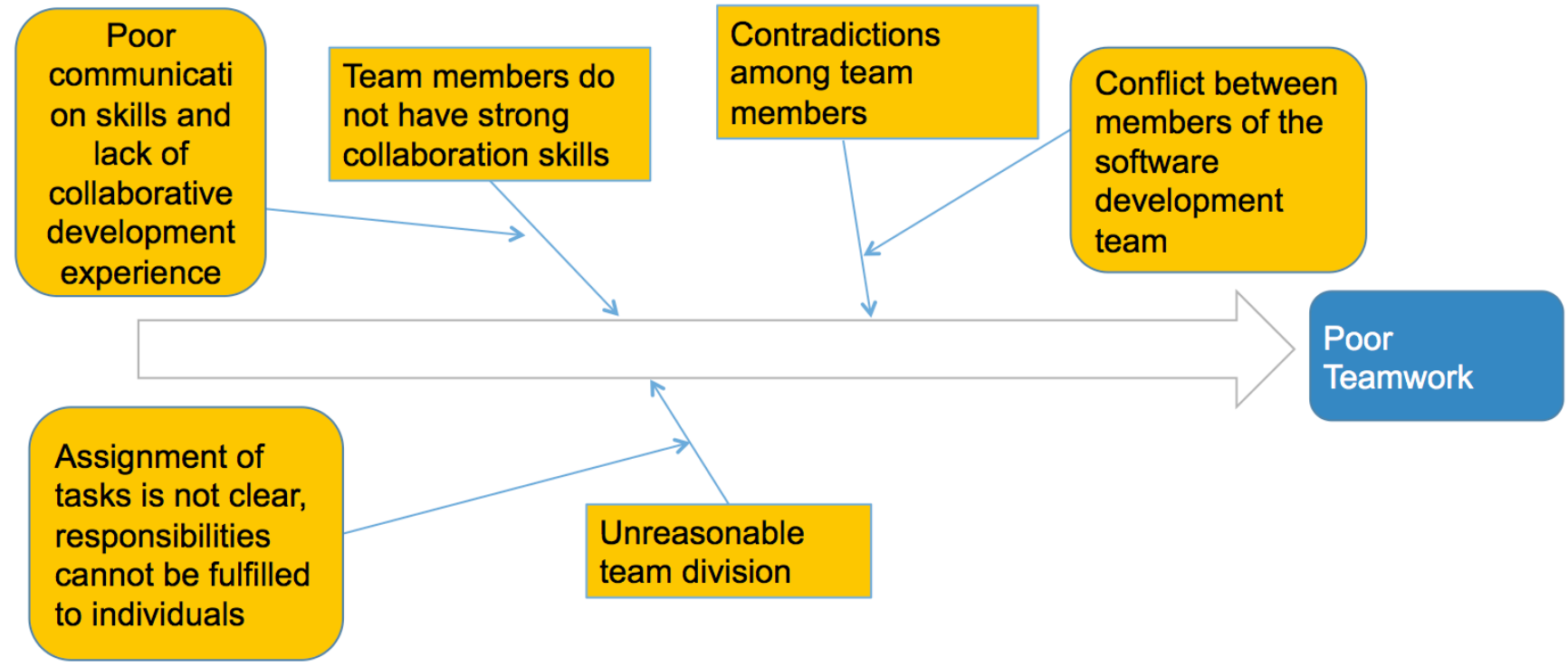
1 : A project change control committee (CCB) was established to determine the changes to be accepted. The CCB is composed of multiple parties involved in the project, including the decision-makers of the user and the developer.

2 : Prioritize new requirements that have already been accepted and implement them in batches. Develop a simple and effective change control process and document it.

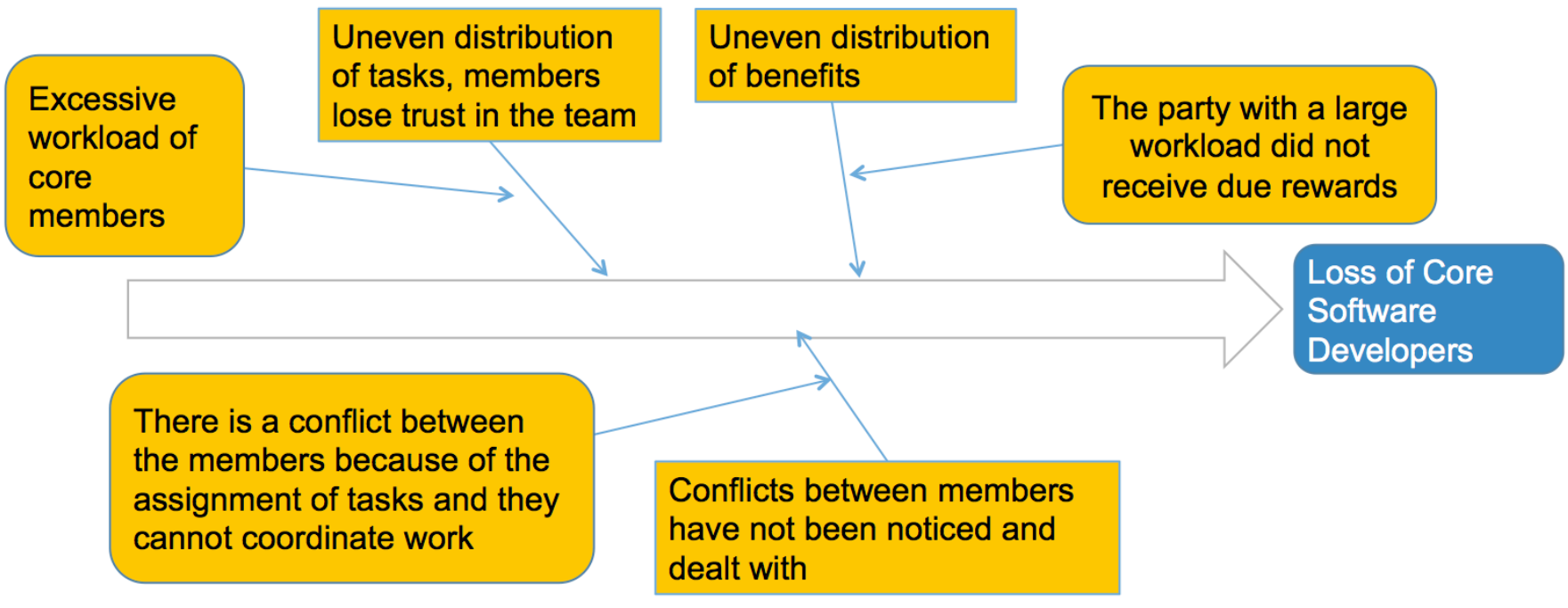


Solutions :

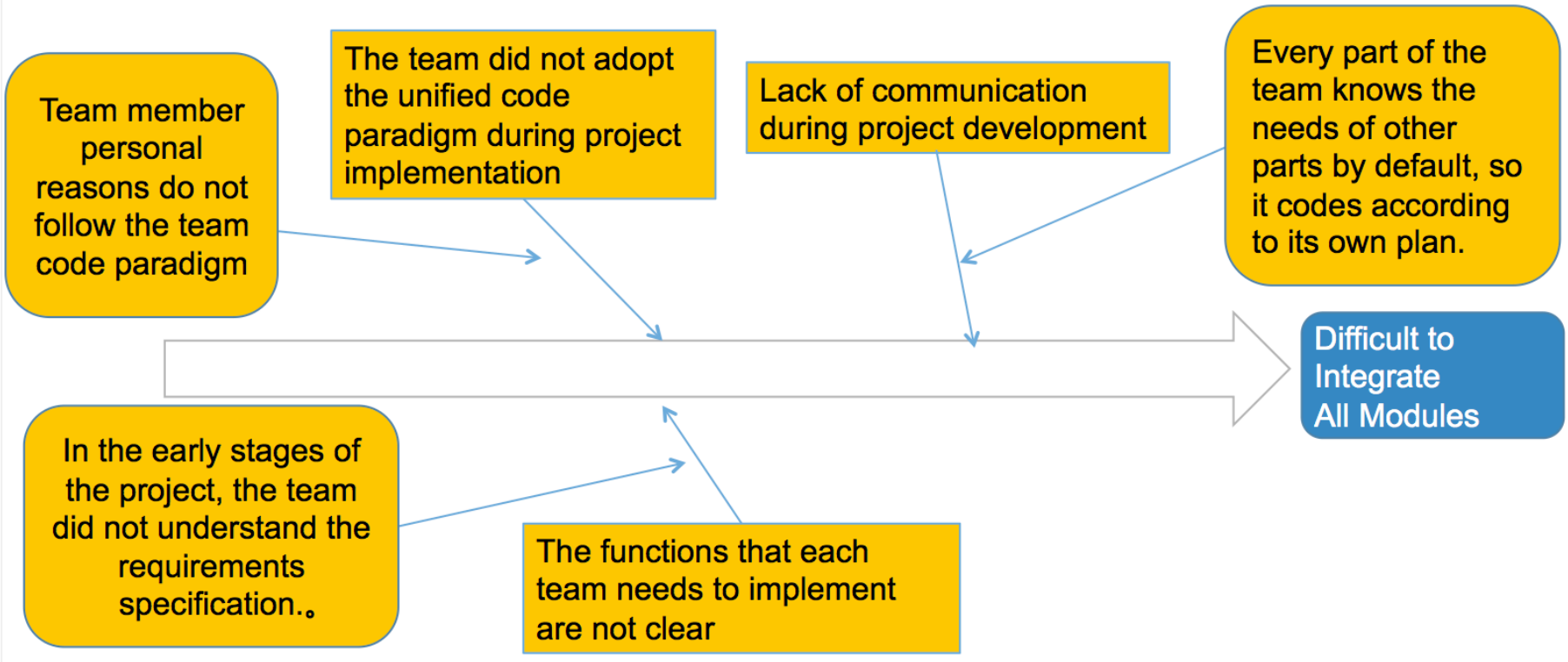
1. Rearrange the project budget after fully investigating project funding
2. Delineate spending priorities and prioritize high-level and more important spending requirements



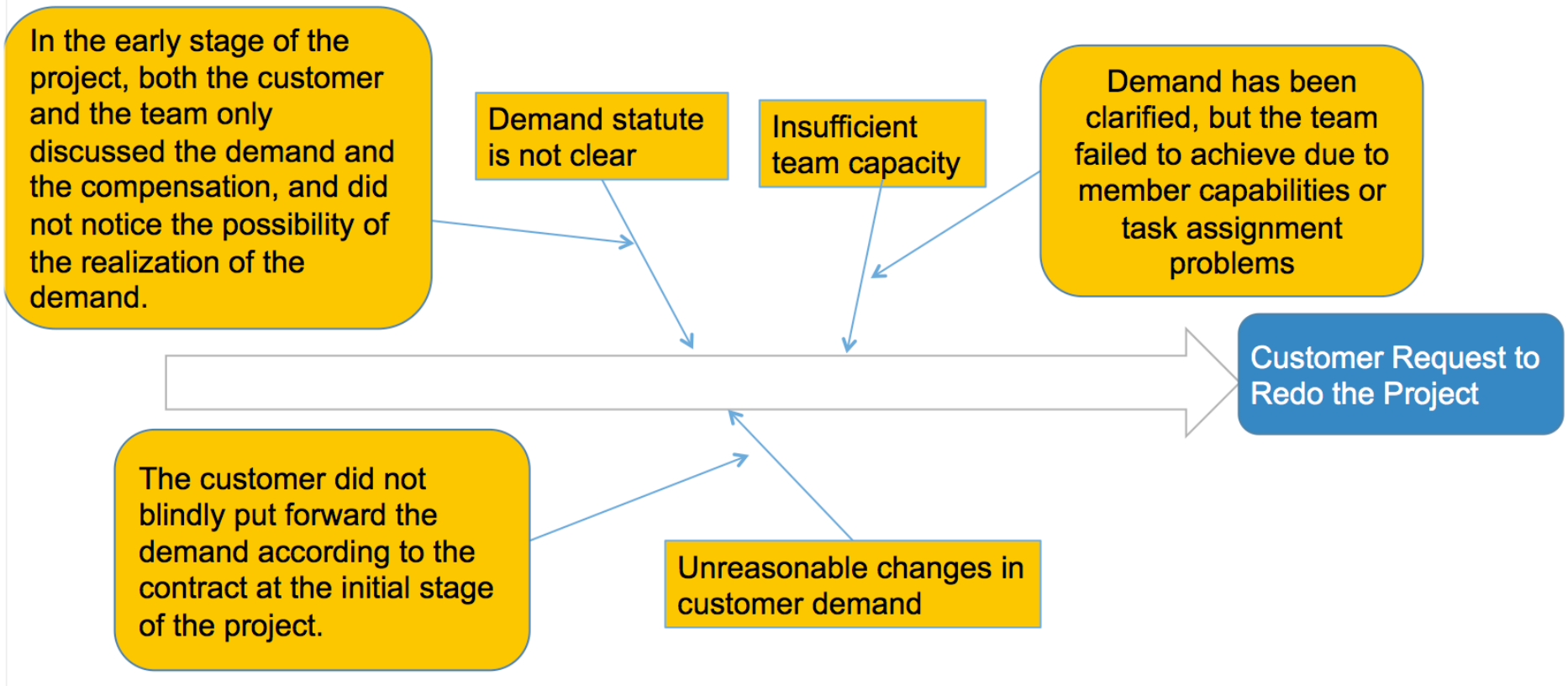
Solutions：
1: The company conducts training on communication skills, cooperation methods, etc., organizes interesting team building activities, and regularly reviews team atmosphere and team image.
2: Analyze the capabilities and advantages of each member and, on this basis, rationalize the division of labor



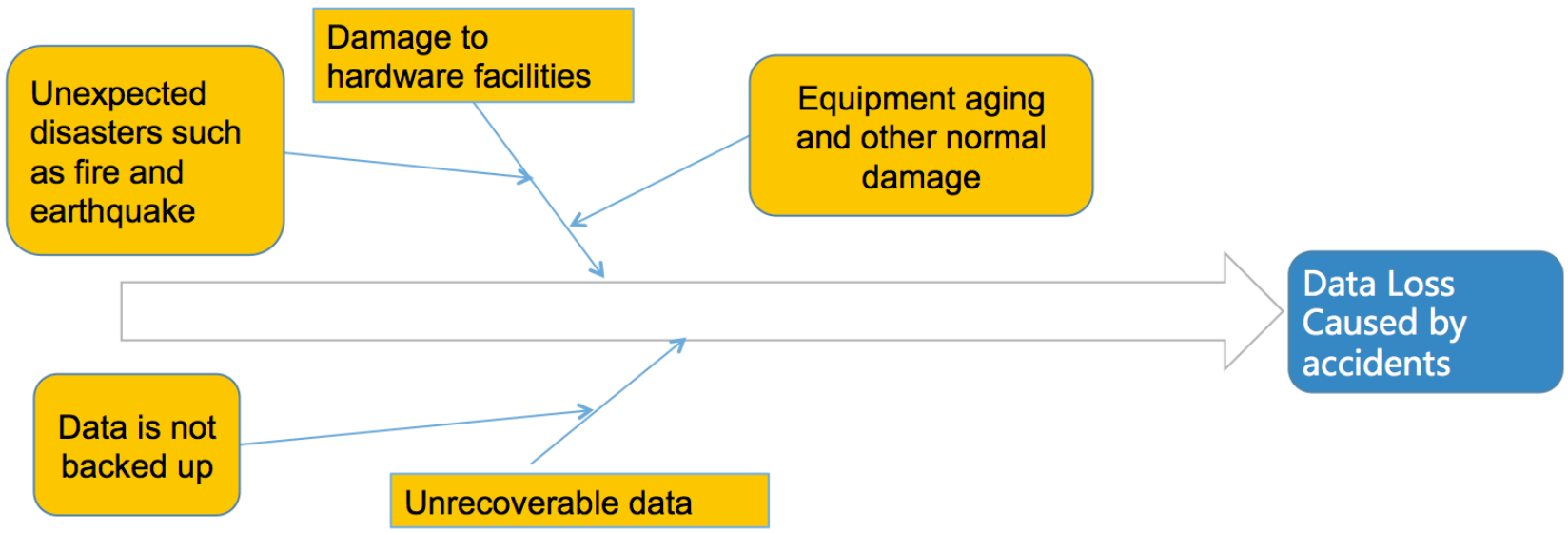
Solutions：
1. Optimize team structure to ensure that every member contributes
2. Distribute workload and benefits reasonably, and reasonably consider employees' compensation according to their contribution to the entire project



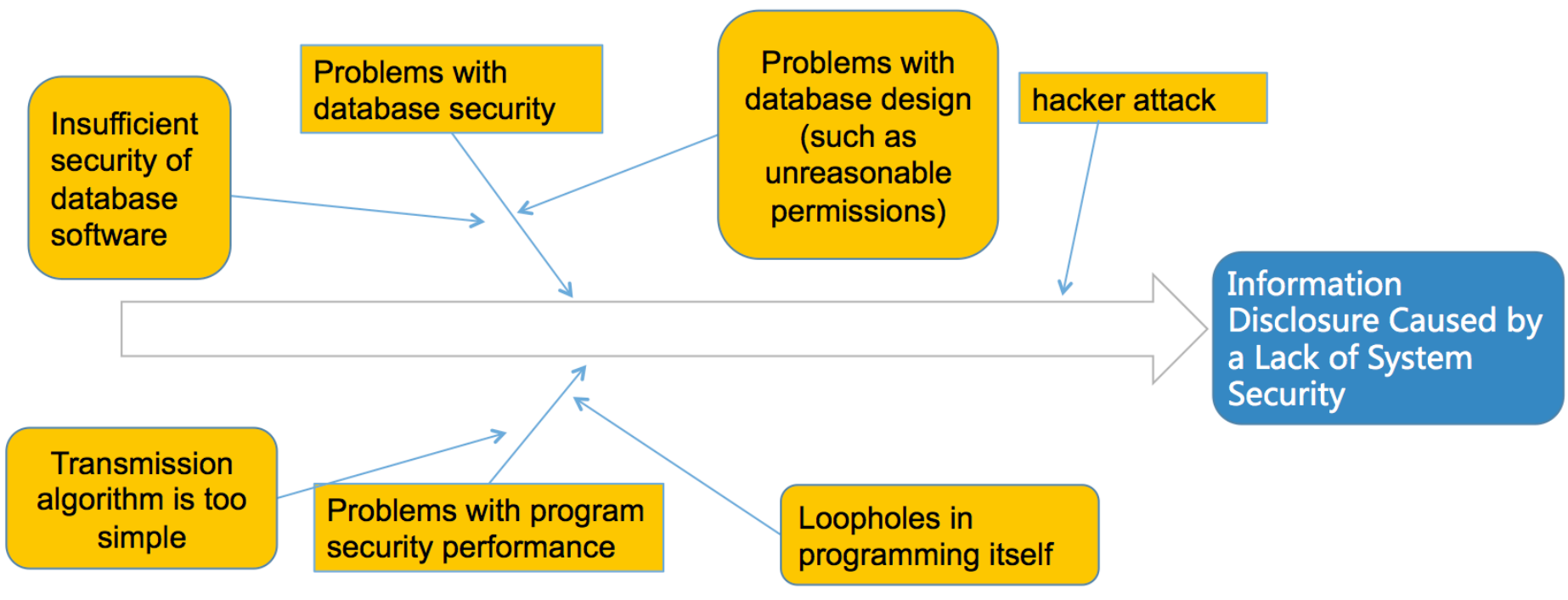
Solutions：
1. Based on the functions that have been implemented in various parts of the current team, the division of work can be rationalized and the responsibility boundary can be clearly defined.
2. Reconstruct the project according to the requirements that are already clearly defined.



Solutions：
1. According to the possibility of reconstruction of the project to decide whether to reconstruct, if reconstructed, re-negotiate with the customer.
2. Identify new requirements put forward by customers and satisfy them as much as possible by adding functions on the basis of the original ones.

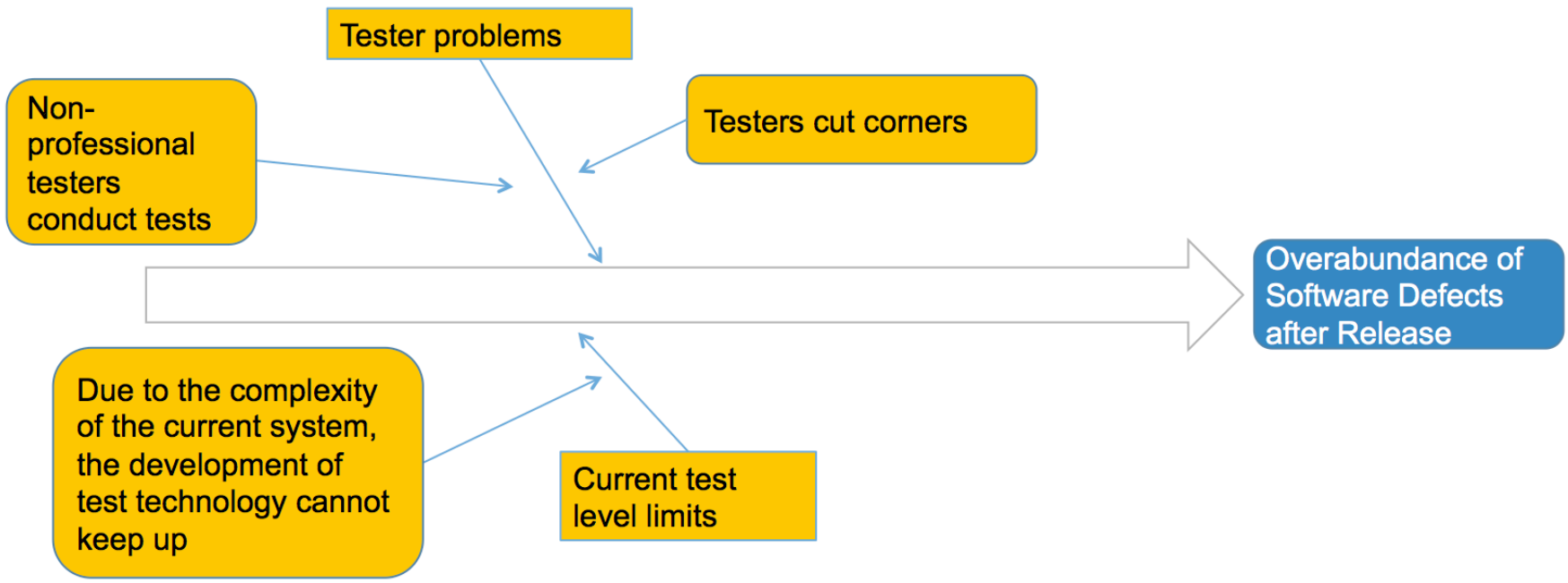


Solutions：
1：Hardware equipment is placed in a relatively safe area (small natural disasters occur), and fire prevention facilities are complete.
2：Multiple backups of data

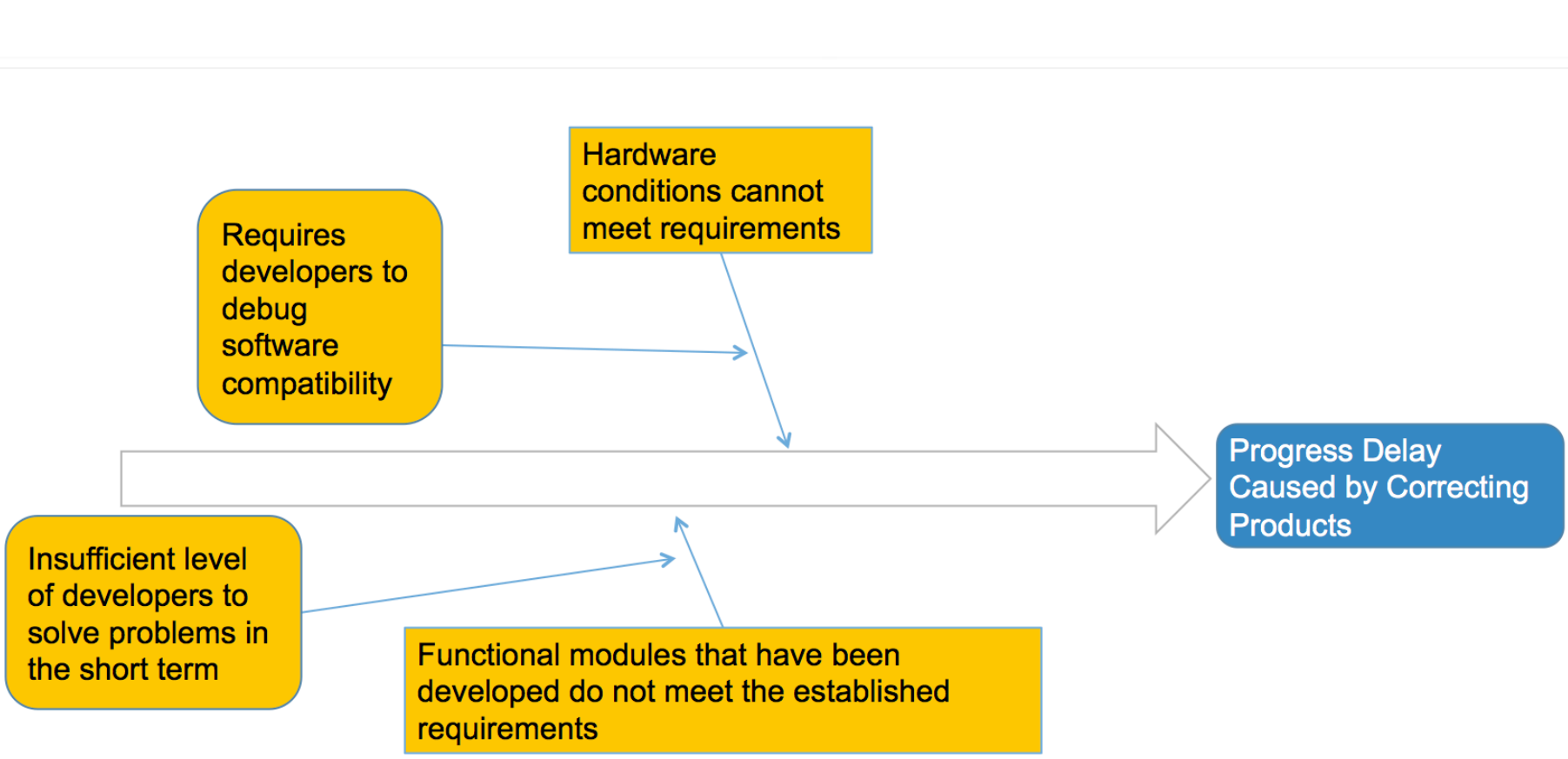


Solutions：
1：The transmission process uses a valid encryption algorithm
2：Strict white box testing of the program
3：Reasonably design the database and use better database software.

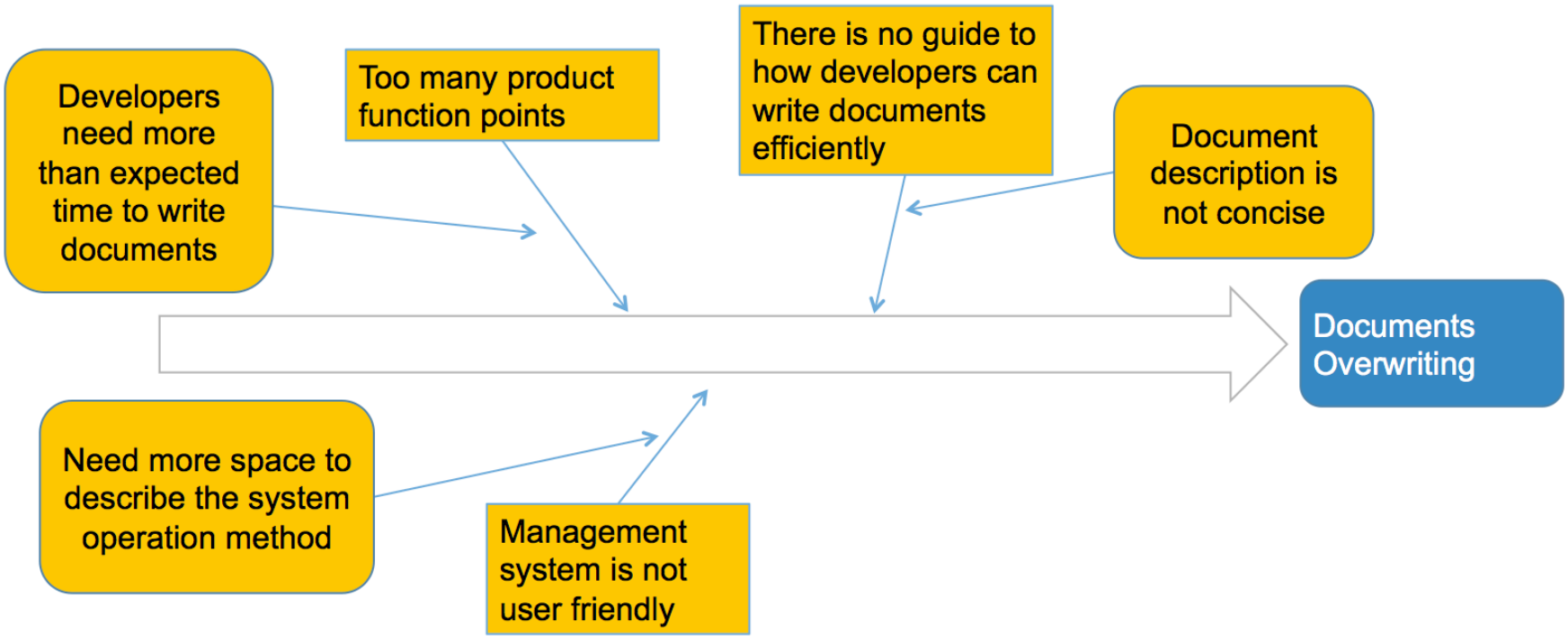
[无标题]



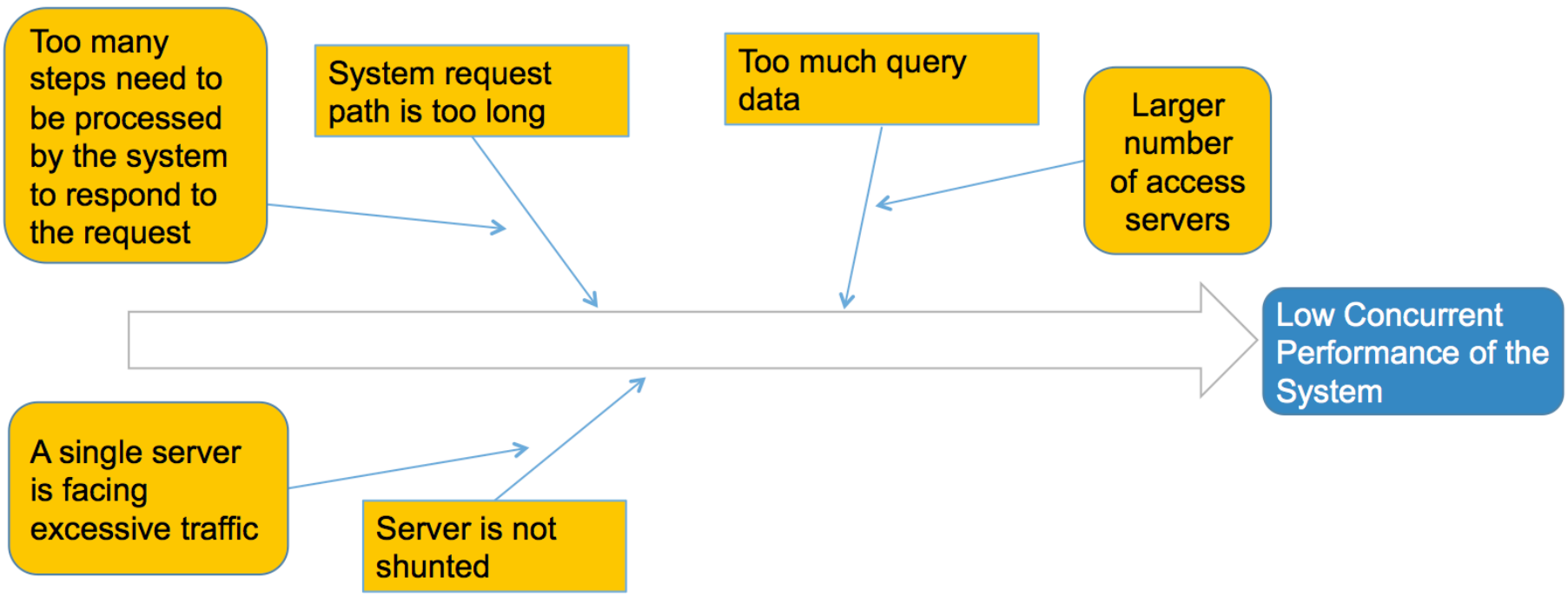
Solution：
1. Hire higher level software test engineers who are capable of effectively managing the testing process



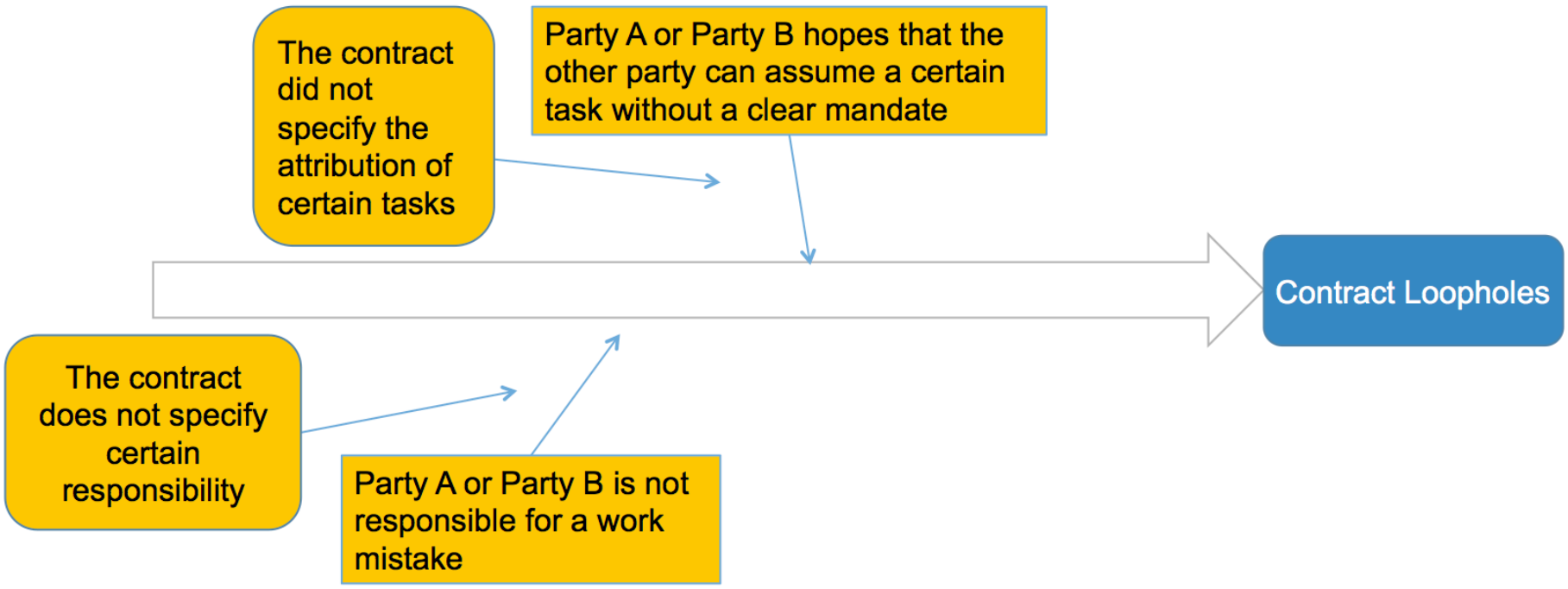
Solutions：
1：Hiring a higher level of developer
2. Spending money to purchase higher-level hardware devices
三. Use agile development to complete the task in a short time



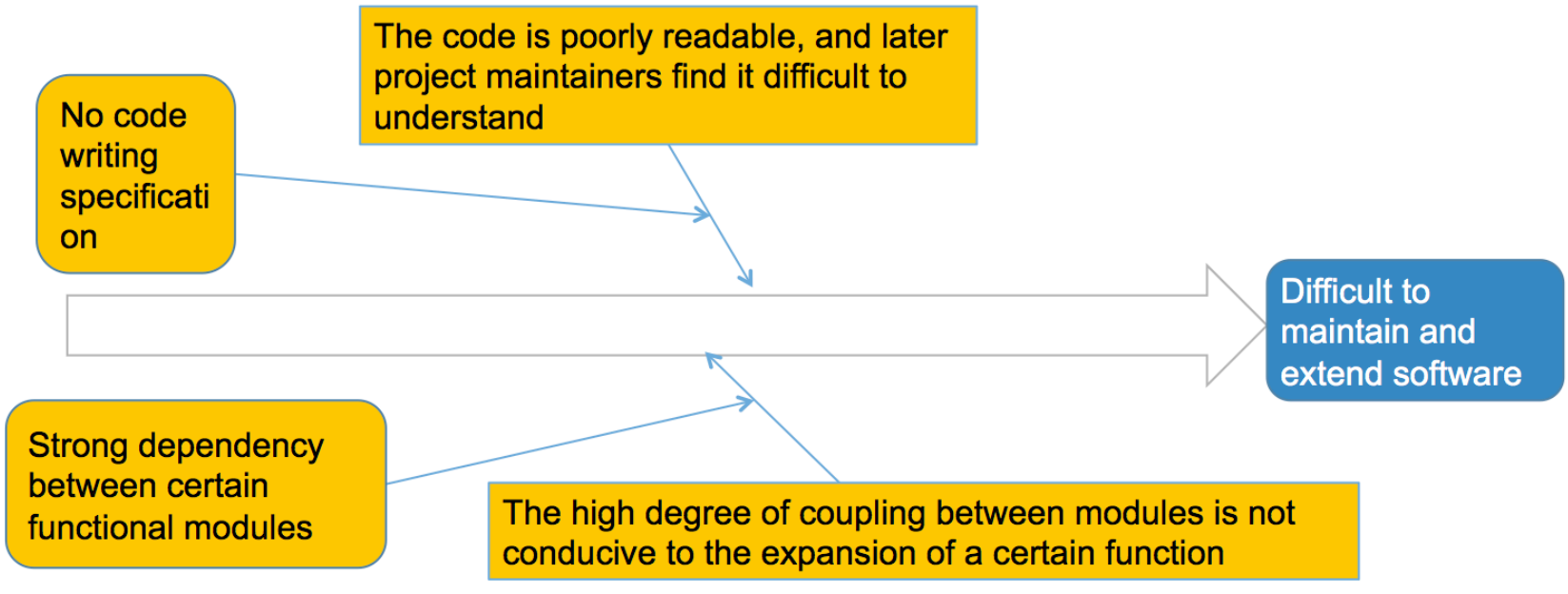
Solutions :
1. Simplify system functions, making the system easier for users to operate
2. Instruct developers how to write documents efficiently



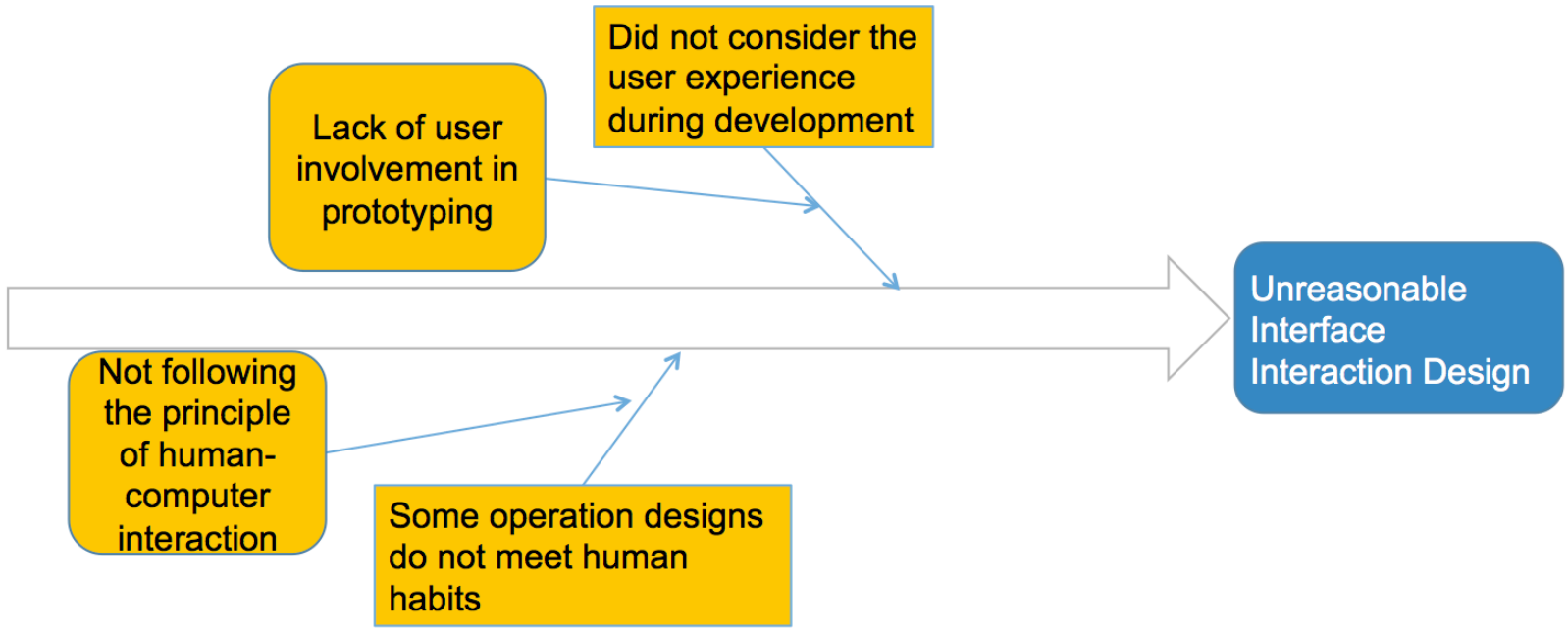
Solutions :
1. Use a distributed cache database
2. Indexes in the database reduce search times



Solutions：
1. Assignment of consultations between the parties
2. Make up detailed contract
3. Seek help from the court or relevant authorities, arbitration



Solutions：
1. Supplementary code writing specification, specifications need to expand the code
2. Refactor some modules



Solutions：
Re-test the user experience, evaluate the loss, and perform interface redo.

Record and Documentation

Then, referring to the audit review method and risk watch list method, a series of tables were drawn as a regular recording tool for risk monitoring. As risks are not all avoidable, we will fully absorb the experience and lessons of project risk management and record them. In order to better adapt to the development and change of the situation, we will continue to make feedback on the existing risk management in the process of risk monitoring in order to adjust the strategy in time. After the occurrence of risk events, it must be archived and reevaluated, because the risk response means will change the probability and adverse effects of the identified risk. After reordering the risk order, we will evaluate it again to make risk management more refined. I should also record and archive the risks that have not occurred, and eliminate them in the project risk plan. At the same time, we need to collate, maintain and analyze the data collected and used in the risk management process, and establish a risk database. Using this database can help the risk managers in the whole organization to accumulate risk management experience over time.

Risk Watch List Example

Potential risk area	Risk reduction Solutions	Activity code	Expected completion date	Completion date	Remarks
Changing or Unclear Software Requirements					
Poor Teamwork					
Loss of Core Software Developers					
Budget Cutting					
Low Concurrent Performance of the System					
Information Disclosure					
Contract Loopholes					
Difficult to maintain and extend software					
Progress Delay					

Five Risks List

Five Risks List

Risk Event	Monthly sorting			Risk solving progress
	current month	last month	the numbers of month	

Risk Monitoring Report

Risk Monitoring Report

Monthly Risk Monitoring Report	
Risk type：	Risk report date：
Risk ID：	Risk description：
Risk state： Influences： Probability： Current risk exposure： Initial risk exposure：	
Risk source and response description：	

Description	Responses	Behavior label

Risk change curve：

Summary

Through the risk management of the Sunstate online leasing system development process, we strictly follow the process and method of risk management, which is divided into four steps.

Risk identification

We use fish bone map, SWOT and RBS tools to carry out risk identification. Through the classification methods to help us brainstorm, we finally select the 15 most influential risk as the result of risk identification.

Risk assessment

In the qualitative analysis, we first define the 10 levels of the risk occurrence probability and the impact of risk, and then use the Delphi method to make the group members score for the two properties of 15 risks. Finally, the average value of the two factors of 15 risks is calculated, and the corresponding risk grade is drawn from the multiplication of the two factors so that the probability influence matrix is drew. In the quantitative analysis, we use the decision tree to evaluate the risk quantitatively, analyze the economic losses of taking different measures to response to the 15 risks; and the tornado map output by sensitivity analysis shows the influence of the sensitive factors directly. Finally, the 5 highest priority risks we get are low concurrent performance, difficult software maintenance and expansion, poor teamwork, changing or not clear demand, and overabundance of defects after software release.

Risk planning

Referring to historical cases, consulting information and combining the related knowledge and experience of software engineering, software testing, and other disciplines, we set up

coping strategies for the 5 highest priority risks selected by the risk assessment process. Why, How, What, Who, When and How much were considered respectively.

Risk monitoring

We consider risk causes through causal analysis, so as to identify and measure project risks early in the process of risk monitoring. Then, referring to the audit review method and risk monitoring method, a series of tables were drawn as a regular recording tool for risk monitoring. Finally, some emergency measures for sudden risks are given.

After implementing the whole process of risk management, we looked back and summed up some of the advantages and disadvantages of the previous work as a result of the preliminary review. First of all, in the process of risk identification, we used a lot of methods to identify more comprehensively, but the risk is not specific and detailed. In the risk assessment, we use the Delphi method and adopt the way of multi person scoring and averaging. Therefore, the risk grade of the evaluation should be more accurate. However, in the process of making decision trees, the evaluation of the loss of risk is still lack of evidence, and the value is inevitably made more or less by the experience. At the same time, Risk identification is not specific enough, since some risks covers too many possibilities, which may not be easily taken into consideration when the loss is evaluated. In the process of risk planning, we have a detailed consideration of each risk, and on the basis of consulting a large amount of information, we put forward a number of coping strategies, including the method of risk aversion, how to reduce the loss after the occurrence of the risk, and so on, which can effectively reduce or avoid the economic losses caused by the most influential risks. At the end of the risk monitoring stage, we monitor the existence of the underlying triggers for each risk in the Sunstate online leasing system, and determine the potential possibility of the risk. At the same time, various emergency measures are planned for risks that occur suddenly, and the audit review and risk supervision are used. It can be said that our work is relative comprehensive.

Next, in response to the problems identified in the review, we made improvements in the corresponding steps. First of all, the risk identified is further refined to make it more accurate and specific. Then, a quantitative analysis of the risk after refinement is remade, and the values in the decision trees are judged and updated through group discussion and data access, to make them more reasonable.

List of references

Reference:

[1] Shen Kaitao (2015) Risk Identification.

Abstract: This book starts with the “risk management overview”, and then proceeds from “property risk”, “risk liability”, “personal risk”, “project construction risk” and “risk evolution trend”. With these aspects elaborated and analyzed the identification and management of risks. Finally, from the perspective of business process practice, "insurance brokers and risk management" conducted a comprehensive summary of the risk management for the insurance brokers.

[2] Marvin Rausand (2011) Risk Assessment: Theory, Methods, and Applications.

Abstract: The book is presented in three succinct sections that discuss the key theory, methods, and applications of risk assessment. The first section lays the groundwork for the topic by introducing laws, regulations, and concepts related to risk assessment, along with in-depth discussion of related topics including random hazards and deliberate threats; threat taxonomies; how to measure and evaluate risk; individual and societal risk metrics; main approaches to setting risk acceptance criteria; the role of risk analysis in safety/risk management; accidents and accident models; and data and data sources for risk analysis. Next, the author treats the key methods and tools for risk analysis. Each method is treated in the same comprehensive format: introduction; objectives and application; method description (qualitative and quantitative); resources required; and pros and cons. Flow sheet accompanies each method, illustrating the various steps and the required input to each step. Additional topics of coverage in this section include coverage of hazard identification, causal analysis, barrier analysis, and uncertainty and sensitivity. A final section focuses on applications of risk analysis techniques and discusses process hazard analysis; risk and vulnerability analysis; environmental risk analysis; major accident risk analysis; product/machinery risk analysis; and formal safety assessment/maritime risk.

[3] Fan Daojin, Chen WeiKe (2010) Risk Management Theory & Tool.

Abstract: Through the introduction of a large number of risk management theories and tools, this book systematically and exhaustively describes the emerging discipline of risk management. The chapter introduces the concept of risk and risk management and the goals and processes of risk management. Chapter 2 introduces the concepts, principles, characteristics, and processes of risk identification. Chapter 3 introduces expert investigation methods, safety checklist methods, and scenario analysis. The specific methods of risk identification such as method and fault tree analysis; Chapter IV introduces the concept, principles and process of risk assessment; Chapter 5 introduces the specific methods of deterministic risk estimation, random risk estimation and uncertain risk estimation. Chapter 6 introduces the specific methods of risk assessment such as AHP, fuzzy comprehensive evaluation, Monte Carlo, etc. Chapter 7 introduces the concepts, characteristics, and principles of risk management decision-making; Chapter 8 introduces loss expectations analysis. Methods for risk management decision making, such as the Expected Value Analysis method; Chapter 9, the necessity of risk monitoring, the determination of monitoring opportunities, and the basis and principles for risk monitoring; Chapter X describes the histogram, causal analysis chart, and Retard diagrams and other risk monitoring tools.

Contributions of team members

Xiaoqing Sun (1552683)

Team organization, risk identification, slides, check the document

Tianyuan Luo (1552754)

Risk assessment, modify and supplement the document, translation

Xiaofeng Xu (1552766)

Risk response plan, presentation, translation, check the document

Yehao Wei (1552695)

Risk monitoring and control, translation, check the document

Huipeng Chen (1552692)

Introduction, summary, group discussion support, check the document