Regarding technical debt, students who are doing software development should be familiar with the following scenarios:

In order to dare to progress the project, detailed design, unit testing and other processes will not be written, and will be supplemented later.

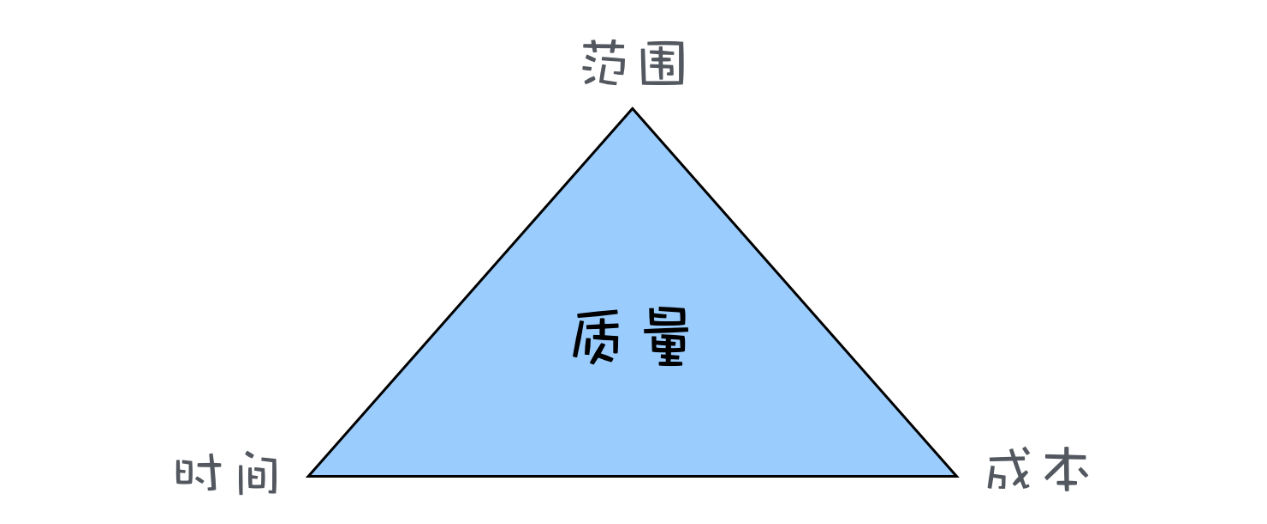
The demand has changed a lot. The original architecture design can't meet the new requirements, but I don't want to move the architecture, so I bypass the architecture design and add new code.

Old system, no documentation, no comments, difficult maintenance

As above, if we don't correct it in time, we will have more and more debts, which will lead to bloated code and inefficient system. We can use the term technical debt to describe the quality of the system.

 So what is technical debt?

  We know the triangle diagram, time, cost and scope of software quality when we are studying project management, as shown below:



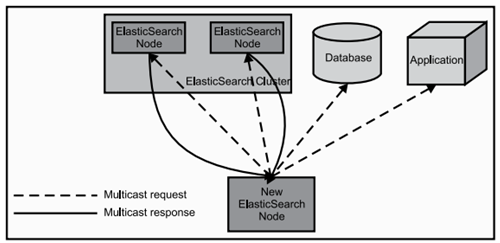
 Why is the quality placed in the middle of the triangle? Because quality is the result of the balance of the other three factors. For example, the scope is not reduced, the cost is not increased, and you want to save time and take shortcuts. Obviously, it will affect the quality. This quality is not only the product quality, but also the quality of the structure and the quality of the code. The overdraft of our quality is a kind of debt, and technical debt is the overdraft of the quality of the architecture and the quality of the code in the software project. Therefore, the technical debt is interest-bearing, and the interest on the debt is the additional time cost when the software is newly modified. Of course, technical debt is not necessarily bad. In software projects, technical debts are often deliberately owed, the development speed of short-term development is increased, software is quickly launched, and the market is seized. There are also models like rapid development, which owe technical debt. Quick verification of the method, even if the verification is not feasible, the technical debt does not need to be repaid.

Then we transfer the problem to elasticsearch. First, let's take a look at the basic content of elasticsearch.

 Elasticsearch is an open source, highly scalable distributed full-text search engine that can store and retrieve data in near real-time. It is very scalable and can be extended to hundreds of servers to process PB-level data. Elasticsearch also uses Java to develop and use Lucene as its core to implement all indexing and search functions, but its purpose is to hide the complexity of Lucene through a simple RESTful API, making full-text search simple.

ES mainly solves three problems: 1) Retrieving relevant data; 2) Returning statistical results; 3) Retrieving faster.

When ElasticSearch's node is started, it uses multicast (or unicast if the user changes the configuration) to find and connect to other nodes in the cluster. This process is shown below:



ES has five core concepts, they are:

 1) Cluster: ES can be used as a standalone single search server. However, to handle large data sets and achieve fault tolerance and high availability, ES can run on many servers that work together. The collection of these servers is called a cluster.

 2) Node: Each server forming a cluster is called a node.

 3) Shard: When there are a large number of documents, a node may not be enough due to memory limitations, insufficient disk processing capabilities, and insufficient response to client requests. In this case, the data can be divided into smaller pieces. Each shard is placed on a different server. When the index you query is distributed across multiple shards, the ES sends the query to each relevant shard and combines the results, and the application does not know the existence of the shard.

4) Replia: To improve query throughput or achieve high availability, you can use a fragmented copy. A copy is an exact copy of a slice, and each slice can have zero or more copies. There can be many identical shards in the ES, one of which is selected to change the indexing operation. This special shard is called the main shard.

 5) Full-text search. Full-text search is to index an article, which can be searched by keyword, similar to the like statement in mysql. Full-text indexing is to segment the content according to the meaning of the word, and then create an index separately.