ADR for Computer Software Transactions
- The Types, Causes and Measures for Prevention of Disputes Involving Software Development -

Kanichi Oishi
Mediation Commissioner
Civil Cases Department 22
Tokyo District Court

1. Introduction

IT-related technology is advancing remarkably but since the knowledge of the average person about the utilized technology is at a low level, the actual situation today is that along with the growing use of computers we see an increase in cases developing from litigation arising from disputes between providers of IT technology and users.

In these disputes, typically the customer, desiring but not expert in information management technology, just winds up leaving the development to the software development company (the "software house") and entrusts it with the building of the system, and the software house commences the development without really knowing the details of the customer´s business, resulting in a completed system that does not satisfy the customer´s intentions. It is also now the situation that there are many cases arising from the fact that the software house, which started up as a venture business and basically is unskilled at business, progresses with development after entering into a contract that is inappropriate to the actual situation, this leading to problems afterwards.

Consequently, here I will provide a summary of litigation cases involving software development that went through mediation in the Tokyo District Court, analyze the causes of the disputes and introduce policies about and ways to prevent such disputes.

2. Types of Disputes Involving Software Transactions

The following are common types of disputes involving software deals:

- Disputes involving the quality of the delivered software;
- Disputes involving the time for delivery;
- Disputes involving the management speed of the system.

1) Disputes Involving the Quality of the Delivered System

There are many cases where lawsuits are based on claims for payment of the development price, payment of which was refused because the system that was worked on and delivered by the receiver of the order (the software house), based on a commission of work from the user of the system (the customer), was not completed in conformance with the intentions of the customer, or on claims for return of the development price where it has already been paid.

When the allegations of both parties are analyzed it can be seen that many of the cases that became lawsuits stemmed from the fact that the completed system was different from the customer´s expectations, since development was started without basically ascertaining what should be developed.

The first cause we encounter is that the specifications have not been sufficiently confirmed to see what should be made, and how. The user basically thinks that since the software house is a computer "pro", it can naturally come up with a system that will meet the user's intentions, and it is noteworthy how many cases stem from a customer placing an order without an understanding of the information management mechanism. On the other hand, we can see cases where the recipient of the order, the software developer, starts out with only meager knowledge of the customer´s business and commences development by thinking that the details of the order can be interpreted as the developer itself thinks fit, and as a result the developed system is different from what the customer sought. It is necessary that before commencing development the software developer prepare design drawings upon receipt from the customer of the required specifications and that both parties confirm that the design satisfies those specifications, and that the parties enter into a development contract specifying the development price, the time for delivery, and so on.

Secondly, there are many disputes to the tune of, "I said that" and "No, I didn't hear you say that", caused by the details of the arrangements made by the customer and the receiver of the order not being reduced to a writing. The cases that particularly stand out are those in which the contents of arrangements made by telephone are not preserved in the form of minutes. It is necessary to put the minutes of the arrangements in writings that both parties then confirm. Especially with computer software, which is recognized to be an intangible thing, in order to avoid disputes it is important to put in writing the estimates, contracts, minutes, memoranda, required specifications, basic designs, test plans, test results, and so forth.

Thirdly, there are cases arising from quality issues, when the completed software is delivered without having undergone a sufficient confirmation of its operation. In particular, many cases that stand out stem from issues concerning how to deal with major defects that are discovered after software that has been delivered is given a "seal of approval" without an actual inspection.

The testing of the created software should include not only methods for accurately checking errors in the developed program and correcting them; in addition, confirmation of the completion of the development of the commissioned software is an important process. That is to say, as a result of this test the resulting product is delivered to the customer and it becomes possible to invoice the development price, so it is an important process of confirmation. One thing that can often be seen in disputes is the case where a test is carried out but only in a desultory manner, and the case where the party that has received the order passes it on to a subcontractor and does not conduct the confirmation process itself. Especially important are coupling tests and general tests (tests of management in exceptional cases; testing of limits, etc.) in which it is determined if the conditions for completion of the software based on the required specifications have been met.

Often seen are cases involving disputes concerning the inspection methods of the customer as to the developed software. As an example of when the inspection becomes a matter of dispute there are cases where test approvals are stamped without the test being conducted and later major defects are discovered, and cases related to a declaration of nonconformity based on acceptance tests that differ from the required specifications.

2) Disputes Involving the Time of Delivery

There are many cases of claims for damages arising from obstacles to planned business due to commissioned software not being completed by the promised delivery date.

The customer complains, "The item I commissioned was not finished by the promised delivery date", or "As a result of a request for remedial work, since the system that was intended was not completed, the delivery date was greatly delayed or the work was not finished." In response, the party who received the order counter-argues that, "In the midst of development there were frequent changes to the specifications and since remedial work was thus required each time the development could not progress as planned." Or, "It was hard to decide on the specs, so the time period for development couldn't be decided." Or, "The product was made in accordance with the specs, but the product must be re-worked due to the strong demands of the customer." Also, there are cases with disputes arising from the claim, "Since the planned quantity of production was greatly exceeded due to the continual changes in the specs, we requested a revision of the development price, but this was not accepted."

In the case of an information management system, the proportion of the development costs attributable to personnel costs is extremely high. For that reason, if the time for delivery is not observed as planned, it is inevitable that losses will be incurred. The costs arising from changes in the specifications have the nature of growing in accordance with the progress in the process of development. Accordingly, in the software industry it is said that failure to appropriately deal with changes in specifications midway through development is fatal.

In recent software development there has been a growth trend in disputes about system development utilizing the "client server" format. With this format, there are many cases of the development of information management using "prototype technology" (see Chart 1).

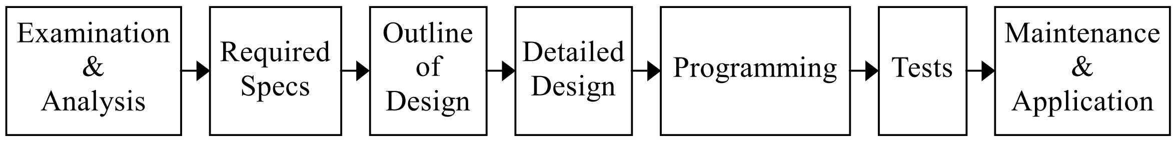
With "prototype technology", trial versions of the system and the program are made without preparing required specifications and basic design drawings in complete form. Instead, it is a method in which the system is assembled while solidifying the specifications by showing them to the customer on a computer screen. This is an easy way for a customer who is not an expert in computer systems to gain understanding, and it is an effective way for the developer as well to understand the customer's requested specifications. So the number of software houses using this method is growing.

Conversely, however, the number of cases is increasing in which what sparks off the dispute is that specifications become difficult to conclusively decide, the time period for development is dragged out and the work is not finished within the contractual period for delivery. Moreover, it is necessary for the software house to know that there is a risk that the undertaking will not be profitable because the number of units to be produced is increased.

Since the time of delivery is an important factor for the customer, the software house must heed the fact that it has a duty to manage the work process by paying strict attention to the date for delivery no matter what.

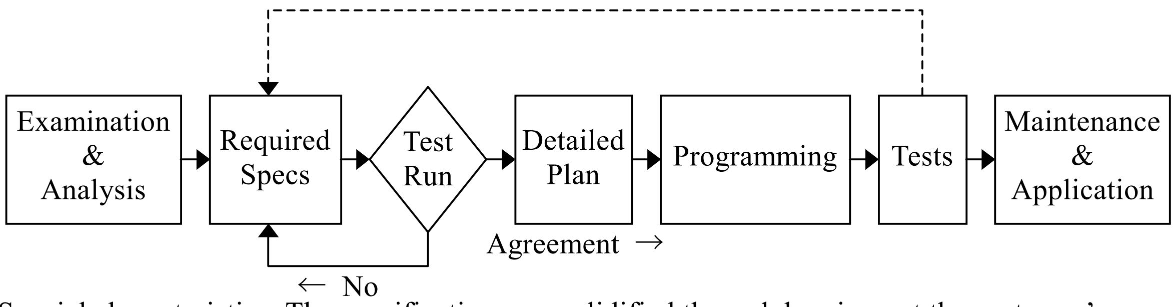
Chart 1. Software Development Technologies:Comparison of "Waterfall Method" and "Prototype Method"

(1) Waterfall Method (basic form; mainly for large-scale and mid-scale systems)



Special characteristics: the system is produced after first confirming the required specifications and so on from the beginning (risk is minimized).

(2) Prototype Method (mainly for personal computer systems)



Special characteristics: The specifications are solidified through hearing out the customer's intentions while showing the customer the images on the computer screen (risk is greater as to an increase in the amount of work and the time of the development period).

The cause for this, in a word, is attributable to unpreparedness in the "project management" of the system developer. In the case of software development, this means that the correct progression of "process management" is an extremely important element. Whether or not a system having the functions stipulated in the documents for the required specifications can be completed by the promised delivery date depends on the capabilities of the project manager who manages the development project.

In addition, the project manager who undertakes this responsibility for the developer has the important obligation of endeavoring to ensure that the developer does not stray from a mutual understanding of the customer's intent. Customers generally have only sparse knowledge about computers, so in order for a developer, who has only a limited understanding about the customer's business, to produce a system meeting the customer's desires, it is the duty of the project manager to devise a way for the two parties to come to a mutual understanding of their aims.

One cause of the revision of specifications midway through development can be seen to be the case attributable to the way in which information is collected as to the desires of the customer's personnel who are concerned with the project. There are also many cases where the cause of this is the lack of management of the project by the customer.

In order for the cause of the dispute not to be attributable to the side that is commissioning the software development, the capacities of the manager who will carry out the project must be looked into. The duties of the customer's project manager will include clarifying the required specifications by adjusting internal plans among the relevant persons in his company and promptly conducting inspections to see if the competed condition of the delivered software meets the required specifications.

3) Disputes Involving Management Capabilities of the Constructed System

When the delivered system is shifted into its actually intended work, there are many cases of disputes involving difficulties that appear due to such things as slowness in the speed of management and abnormal stoppages so that business could not be done.

In this type of dispute, the majority of the problems that can be seen develop when no problems are found in the acceptance inspection but are discovered when the system passes on to the operation phase. There are cases that go to litigation for a return of the development price and compensation for damages because even though a request was made to the software developer for improvement of the software, that was not done at all, or because of a complaint from the software house that it was necessary to re-make the software by reason of the amount of data that was estimated at the time of development being greatly exceeded in actuality and a request for reimbursement of the expenses for making the improvement, the customer often alleges that it was the software house that made too optimistic an estimate of the amount of data, so there should be a return of the development amount and compensation for damages.

The core of the problem is caused by mistakes in the estimates of the amount of data to be managed or superficiality in the system tests and acceptance tests. With respect to these problems there are times when they can be solved by improvement of the constructed software itself and times when the problem is about the hardware that is used. In any case, a great amount of time and expenses are needed for the improvement. Disputes develop as to whether the liability for the error in computing the estimate of the amount of data to be managed is attributable to the customer or to the developer. There are a variety of cases. In one type, since the customer was not conscious that the speed of management is different depending on the quantity of data, it did not indicate the quantity in the required specifications and neither did the software house pay heed to that. In other cases, the estimate by the customer was in error. In any event, it is difficult to state unconditionally that it was the customer's error alone. The software house is in the position of a pro, and it is necessary to consider that it has an obligation to make a forecast as to how the management capability will change according to the amount of data and, after explaining that to the customer, to confirm with the customer the required specifications documents.

Furthermore, there are many users who think that an information management system on a computer can be used forever once it is installed. Measures cannot be devised for system errors in operation and abnormal stoppages due to ordinarily-occurring errors in operation, input of abnormal data and the like if there is a “hands-off” philosophy after the system has been introduced. In order to prevent these problems it is necessary to pay attention to the fact that an in-company operation and maintenance system must be organized and appropriate measures must be devised to cope with abnormalities each time they occur.

3. Measures for the Prevention of the Occurrence of Disputes

Certain basic matters must be arranged in advance in order to prevent disputes involving computer software development.

1) Collaboration in System Development by the Customer and the Receiver of the Order

In order to carry out system development efficiently, it is vital that the work stands on the basic recognition that “the development will be done jointly by the customer and the receiver of the work order, with each of them having to bear responsibility for its respective role.” For this purpose, it will be necessary for each of them to appoint a project manager who will manage the execution of the project, and to perform the role that was allotted to each of them through careful liaison. As part of the qualifications of the project managers, in addition to their having authority and executive faculty for the project within their respective companies, it is indispensable that they have a basic knowledge of each other’s business. The customer should have basic knowledge about information management and the developer should have basic knowledge of the business for which it must build software. For example, if the two parties are “speaking a different language”, since neither will be able to understand what the other is saying they will not be able to come to a meeting of the minds as to intent.

The first duty is for the customer to clearly convey to the developer the specifications of what it wants built (the required specifications). The developer must prepare basic design drawings of a system design that reflects a correct understanding of the required specifications furnished by the customer, and then, after getting the confirmation of the customer thereto, it is necessary to proceed to a work contract. If development winds up getting started without precisely clarifying the specifications, this can become the wellspring for disputes, as mentioned above.

So that disputes do not occur, all arrangements of the two parties should be reduced to writing, with both sides retaining copies.

The project managers should bear the following responsibilities.

On the customer’s side:

- carry out with exactitude necessary adjustments with persons in the company who will be relevant to the system that will be built;
- furnish the developer with required specifications that clearly indicate the details of the system to be built, or prepare such specifications through consultations with the developer;
- clearly express to the developer the plan for introduction of the system, or decide and manage this through consultations with the developer;

- clarify the allotment of roles of the two parties, from system development through working operation;
- promptly conduct an acceptance test as to whether or not the delivered system was produced pursuant to the required specifications;
- establish and carry out a system for operation and maintenance of the system after it is introduced;
- other.

On the developer’s side:

- confirm that there is the necessary skill and knowledge, taking into consideration the client’s intentions;
- jointly with the customer examine the required specifications and prepare basic design drawings, and get the customer’s agreement thereto;
- carry out management of the process in accordance with the development schedule and manage the budget;
- provide proper instructions to and management of the project team (including management of subcontractors when they are used);
- other.

2) Development Process of the Information Management System

Roughly, the development of an information management system consists of the following process.

Table 1. Development Process of an Information Management System, and the Ratio of Each Phase in the Development Process

Process	Contents of Development Process	Ratio of Each Phase
Planning	Examination & analysis of actual conditions	30%
Design	Requirement specs; basic designs; detailed designs	
Production	Making the program	20%
Testing	Separate & coupled tests; system tests	40%
Acceptance	Acceptance tests; switching over to the system	
Operation	Operation & maintenance of the system	

As indicated in Table 1, the phases from deciding on the specifications through confirmation and on to design take up a proportion amounting to 30% of the development process and make up the most important steps. The phase from checking the manufactured system up to the acceptance inspection upon delivery to the customer and on through switching over to the system amounts to 40% of the process. Through the thorough conducting of the tests of the manufactured system there can be a declaration that the developer’s system has been completed, so this is an important phase in the process. Note that production of the program takes up no more than 20% of the development process.

It is necessary for the customer to transmit to the developer in as detailed a writing as possible the terms and conditions for the system it wants to have developed. Table 2 lists the items that should be mentioned in that document.

Table 2. Gist of Makeup of Requirement Specifications

(1) Purpose of systemization	(5) Table of input and output power of information
(2) Applied work of systemization	(6) Table of files
(3) Definition of function of systemization	(7) Development schedule
(4) Definition of system logic	(8) Other (terms and conditions for bidding)

In order to confirm that the system design reflects a complete understanding of the required specifications received from the customer, the receiver of the order should prepare the basic design drawings and commence development after getting the customer's agreement thereto. Table 3 indicates the items that make up the basic design drawings.

Table 3. Items That Make Up the Basic Design Documents

(1) System outline	(2) Design of input and output	(3) Code design	(4) File design
application work flow	screen transition chart	code system	master files
system configuration chart	screen layout	classification system	data files
subsystem definition	specifications of output requirements		total files
development schedule	specifications of output items		file list

Table 4 shows a summary of the tests conducted by the developer itself and the tests carried out by the customer after delivery, in order to confirm the working of the completed system and the conditions of completion.

Table 4. Summary of Tests

Tests carried out by the developer before delivery: - Stand-alone tests: operation tests of each subsystem - Coupled tests: confirmation of operation of the system as a whole by combining all the subsystems - General tests: confirmation of operation with different quantities of data; operation at the time of insertion of abnormal data; operation at the time of abnormal handling, etc.
Tests carried out by the customer after delivery: - Acceptance tests: confirmation of whether or not the finished product was made as stipulated in the required specifications (attendance by the developer is desirable)

- System operation tests: system installation trial run (attendance by the developer is desirable)

Further, the development process for software is being standardized internationally, and in Japan as well Japan Industrial Standards (JIS) have been established corresponding to the international standards. These standards relating to software products and services have provisions concerning the entire life cycle of the software from the time it is conceived on through when it is abandoned, centering on the time from when the order is received until when the software is furnished to the customer. As a result, if the standards for the process for development based on these provisions are observed, and if development proceeds with a careful exchange of information with the customer, disputes should be avoided.

Japan Industrial Standards: Software Lifecycle Process JIS X 0160: 1996
(based on these international standards: ISO/IEC 12207: 1995

4. Conclusion

Above I have presented a summary of the litigation cases involving software development, an analysis of the causes leading to disputes and some measures for preventing disputes. Here is a list of measures to prevent disputes involving software development.

1) Once the developer has received the required specifications from the customer, system development should start only after both sides confirm that the developer has a correct understanding of the specifications, by means of examination of the basic design drawings.
2) There should be a practice of putting in writing the details of the contract, the results of the parties' meetings and arrangements, and so forth.
3) In order for the system development to succeed, the scope of the fielding of problems by both parties should be clarified. Based on an understanding that the development will be joint work, the role of each party's project manager will be a large one.

4) It is important to conduct confirmation tests of the produced system before delivery, to make sure that it operates as per the required specifications.
5) An acceptance inspection test should promptly be conducted on the delivered system pursuant to the required specifications.
6) A system for the operation and maintenance of the system should be established, to be followed after the system is installed.
7) It is desirable that the system development be carried out pursuant to the JIS provisions.