

# Why IT Projects Fail

For the Fifth Audit Seminar of the INTOSAI Standing Committee on IT Audit

Ryoichi Doi (SAI-Japan)

IT projects fail. Although we can say nothing is perfect, there is a difference between projects that fail and minor flaws. For example, developers cannot finish writing each program with complete satisfaction. Although they may see some minor flaws in the program and have new ideas, they have to give up developing the program and just finish. However, projects with minor flaws are completely different from failed IT projects. The following examples of IT failure, taken from *The Annual Audit Report of SAI-Japan* (the Board of Audit of Japan), were submitted to the Diet on November 2006:

- Local governments had independent network systems that connected residents' houses to the government offices. After these governments were merged and the network systems were integrated, the equipment, such as servers, no longer necessary for network management should have been used for other purposes, for example, as Web servers. However, the equipment was left unused.
- The Ministry of Agriculture, Forestry, and Fisheries constructed a system to trace processed foods from production to sales. The system was intended to secure consumers' trust in the safety of the processed foods. But there were few users for the system. Then, system use was ended because the foods did not sell well.
- A system was constructed for managing information about contracts for farming plots. But 5 years after the system began, no data had been input in the system.
- Devices for a fax and telephone system, for simultaneous warning of emergencies at nuclear power plants, had not functioned properly. The devices had been left out of order for 1 year and 9 months. In addition, some devices had never been inspected, despite the fact that the nuclear agency had paid maintenance fees for 7 years and 10 months. Finally, batteries for a power failure had not been changed though the agency had paid the fee.
- The police had invested 900 million yen to pay for introducing and operating 58 sets of the Forged Credit Card Analyzing System. After 3 years, only 200 credit cards had been scanned through the system and registered in the database. The police, however, had seized approximately 41,000 forged cards.
- The government built an electronic record system for patient cases. But part of the system still does not work, and the future of the system is vague because of delay

in software development and loss of developer responsibility for integrating the system.

### **Causes of Project Failure**

IT projects that fail continue to be developed. Often, IT is seen as a magic box. But it can only do what a human being can do. Nevertheless, some tend to envisage an epoch-making function for IT, placing it at the center of an organization. Consequently, especially in the public sector, it seems the failure of IT projects will never be stopped. Is there any way to avoid such failure? One way is avoiding the two main causes of failure: excessively large systems and poor vendor control.

#### Excessively Large Systems

When a project fails, the system is often too large for the developer to control. However, not only vendors, but also program managers in charge of outsourcing, tend to find a large system attractive. Both vendors and program managers want a large system because

- given enough budget, it includes as many functions as possible, as well as those of a smaller system;
- they believe a larger system can be more effective; and
- they believe the importance of an organization is in proportion to the amount of money invested in IT.

But are excessively large systems really effective for the organization and users? The answer is no. This is because such a large system

- includes a number of uncertainties that may result in errors;
- takes time to be completed, but meanwhile the technological, social, or political situation may change resulting in new requirements extending deadlines;
- is more difficult to deal with when unexpected problems--which are hard to predict--arise, such as delay in responding to the user accessing the system;
- may be useless unless stakeholders (program managers, procurement managers, developers, and users) take time to understand critical system specifications; and
- involves vast amounts of money, which enables vendors to gain more profit usually--whether or not the project is successful--but makes it difficult for stakeholders to withdraw funding after investing vast amounts of money.

#### Poor Vendor Control

Another main cause of IT project failure is that vendors are not properly controlled. The IT project goes forward, according to the contract, at the mercy of the vendor. But the vendor is not responsible for the result--success or failure. Therefore, in the end,

- the project does not make progress,
- unnecessary functions are added,

- the system does not take users into account, and
- the completed system is not used.

Often, procurement managers, who are typically in charge of vendor control, cannot catch up with rapid changes in technological trends. They therefore swallow, without digesting, proposals from vendors. In such a situation, it is impossible to realize good vendor control. However, vendors also have a difficult time. They are professional developers of systems, but they are not given the flexibility to develop a given system as they deem appropriate and may not have input into the operation of this system. In addition, vendors, even though they try to do their best, are often not given enough or necessary information to develop a good system.

### Causes of Project Success

To avoid IT projects that fail, remember the time you wrote a program for the first time. At the beginning, you were satisfied just to confirm that the program was running properly:

```
printf( "Hello World!!\n" )
```

Then you became able to write larger programs, step by step, testing incrementally. This method--small systems at first--should be used to develop new systems. At the same time, program managers and procurement managers must communicate with other stakeholders and have a sense of responsibility. In particular, before a project is begun, program managers need to convey their vision of the method to other stakeholders. And procurement managers need to retain control of vendors. Finally, assessment of the IT project is essential to success.

### Small Systems at First

Stakeholders should understand the purpose of the project. Their understanding of system specifications can reduce the risk of failure. They can then authorize developers to develop part of a system or a whole system with functions that are limited to really necessary ones. And they can ascertain whether or not the small system is useful. Even if stakeholders decide to stop the development of this system, the amount of money invested is little, so the damage will be less than developing a large system and abandoning it.

Making a small system at first and then bringing it up to an appropriate size is the most effective way to develop a system. While still small, but completed, the system should be tested, including the checking of the following items: system-response time, maintainability, and affinity between users and the system. Such affinity is likely to be forgotten or difficult to test in the development stage. But after the system starts

operation, the specification requirements become clearer and more appropriate.

Given that a small system can be completed in a short time and at low cost, the lessons learned in developing the system can become the basis for the next system. If you develop a small completed system successfully, stakeholders can understand system functions even if they cannot read human-computer interfaces.

### Good Vendor Control

All stakeholders for a project--including, at a minimum, program managers, procurement managers, developers, and users--should be involved in control of vendors that provide products or services, such as analysis. If stakeholders, especially procurement managers, do not make a correct decision and take an appropriate action, vendor control will never be realized. Consequently, the entire project will collapse. But different stakeholders have different interests:

#### Program Managers

To be able to control vendors, it is necessary that program managers show a clear vision for the project. In addition, they must control procurement managers--especially those giving instructions to the vendor--who are prone to develop a large system from the beginning. Program managers must shore up procurement managers' will, improve their skills, and urge them to continue their efforts. Program managers are typically interested in improving the performance of the IT system, but they often do not look at the contribution of their own performance in such improvement.

#### Procurement Managers

To make the project successful, procurement managers need a strong will, IT skills, and daily effort to effectively oversee and carry out administrative services for the project. But they should also have the ability to (1) to assess and develop the best plan or (2) develop an alternative plan if the ones suggested by the vendors are not adequate. For example, the vendor suggests plan A, which is an ordinary, boring idea; plan B is an idea to decorate plan C; and plan C requires excessive cost to realize. The vendor is trying to induce the procurement manager to agree to plan C, which brings the vendor greater profit. Plan C, however, is not always the best option for the organization. Often, it is exactly the opposite.

#### Developers and Users

Developers make efforts to develop a system that is easy to administer, but they do not like to spend time explaining the design to other stakeholders. Users inside an organization cry for user-friendly systems, but they do not want to participate in system design meetings.

All stakeholders, however, must deal with the fact that vendors pursue maximum profit while providing minimum resources.

#### Stakeholder Assessment

Finally, all stakeholders, as well as system auditors, should take on full responsibility for assessing the project. In particular, when a specific plan is selected, stakeholders should ask for justification from procurement managers. And procurement managers should (1) keep trying to collect as much project information as needed to make an objective assessment of the project and (2) be ready to be accountable to all the stakeholders.

All stakeholders should also call for second opinions from system auditors. It is said that it is better, in the early stage of system development, to join system auditors from the private sector with the team, asking them for opinions. But program managers should show a blueprint of the project to independent experts before they rely on system auditors. Just inviting system auditors cannot ensure project success. Such success can only be realized if system auditors have a strong will to make the project successful. Generally speaking, system auditors give a course correction to some extent, but never say NO when management says GO. Basically, people will not disagree with those who pay them.