System Dynamics: a New Tool for TQM

Shoji Hidaka
NTT DATA Corporation
Shinkasumigaseki Bldg. 18th Foor
3-3-2 Kasumigaseki, Chiyoda-ku
Tokyo 100-0013, Japan
Telephone: +81-3-3506-4631, Facsimile: +81-3-3506-4638
E-mail: hidakas@noanet.nttdata.co.jp

Abstract

This paper discusses roles of System Dynamics in software quality control. In Japan, recently, a new quality project was launched. We changed the title from TQC to TQM focusing more on software/service quality as well as hardware quality. Software quality control, with lots of human factors, is sometimes too complicated for ordinary statistical tools. This is where System Dynamics plays an important role. In a computer system development project, users had been experiencing thousands of troubles every year and they were wondering whether to expand the system in the next year. A statistical study showed the troubles would disappear till the expansion, which nobody was sure. I introduced a more persuasive approach, System Dynamics, to estimate the future situation, involving users of the system, which made them confident about the result. This paper concludes System Dynamics is a quite useful tool for quality control and will greatly contribute to TQM.

1. From TQC to TQM – Japan's New Challenge

TQC (Total Quality Control)

In 1980s, Japan was leading in producing quality goods by the quality control system, TQC. Japanese people learned this statistical technique from American gurus, such as Dr. Deming. QC circle activities were seen everywhere in Japanese organizations and all employees were involved in them. 'Kaizen', or continuous improvement, was already in English dictionaries. However, in 1990s, TQC was studied all over the world and Americans introduced more drastic approach, such as BPR (Business Process Re-engineering) and benchmarking. They are now leading the world quality goods market.

TQM (Total Quality Management)

In 1996, Japan introduced the global standard of TQM in order to restore the reputation of 'the best quality' by changing the vision and the approach.

TQM emphasizes management responsibilities in organizations to assure the high quality. It uses more top-down approach rather than bottom up approach used in TQC. In addition, Japanese firms now focus on software and service quality control as well as hardware quality control. In the current global competition, competitive advantage is from software and service quality.

I think that new tools will be required for this new approach, which has not been well discussed in Japan. Traditional tools are not always suitable for the changing environment. I hope this study can contribute to the Japanese new quality movement to restore the reputation.

2. TQM and System Dynamics

In this section, I would like to discuss how and where we can use System Dynamics as a new tool in TQM activities. In quality management area, simulation/modeling techniques are not yet well used in Japan. However, thanks to the progress in the computer technology both in hardware and software, even managers can use simulation on their desk top today.

Characteristics of System Dynamics

There are several simulation techniques, but System Dynamics has some preferable characteristics for management activities. They are:

- Good feedback theory and accumulation of case studies
- Visual chart to share the vision
- Good computer software

I have been applying this technique to some management area, such as marketing, Business Process Re-engineering, and facility management. From these experiences, I believe System Dynamics can also be applied to the TQM activities that are based on the PDCA (Plan-Do-Check-Action) cycle.

'QC Story'

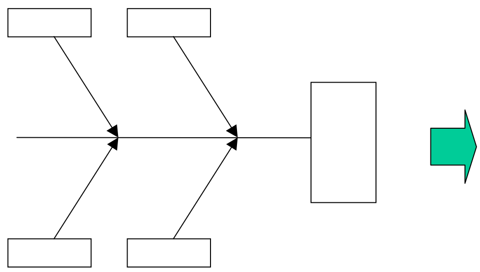
In Japanese quality activities, we use 'QC Story', a thinking framework, to design a problem solving process. It usually contains nine stages:

- Selection of a Theme
- Current Situation Analysis
- Setting the Target
- Cause Analysis
- Solution selection
- Implementation
- Effect analysis
- Prevention of Backsliding
- Insights and Future Direction

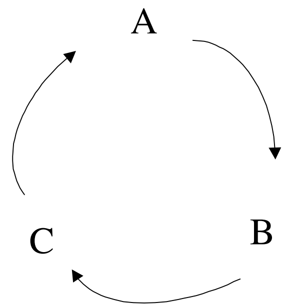
In these stages, System Dynamics can effectively be applied in the three: Current Situation Analysis, Cause Analysis, and Solution Selection. In Current Situation Analysis stage, System Dynamics enable us to analyze future situation with current world structure. In Cause Analysis stage, cyclical relationships will be taken into consideration instead of traditional one-way relationships, which leads to more accurate analysis. In Solution Selection stage, with System Dynamics, managers can analyze the effect of the solutions before they implement them. I will describe these through a case study in the next section.

QC tools and System Dynamics

In TQM activities, several QC tools (known as '7 tools') are used by any employees in Japanese organizations. One of the well-known tools is 'fishbone chart' or 'Ishikawa Diagram'. We have been feeling this fishbone chart has one serious difficulty to understand the world surrounding us. It describes the relationship between cause and effect with an arrow that has only one-way direction. There are lots of relationships around us which can be described as a cycle. In System Dynamics approach, 'Causal Loop Diagram' is used to describe the cyclical cause and effect relationships.



Fishbone Chart



Causal Loop Diagram

I think that introduction of Causal Loop analysis would greatly help us to understand the world and improve the 'quality' of the cause analysis in the TQM activities.

Statistical tools, such as control chart, histogram, regression, are also often used in TQM activities in Japan. They are suitable for hardware quality control which has solid relationships between cause and effect. It is often difficult to use them for software quality control which has many human factors. In System Dynamics approach, there are some good tools, such as graph function, to treat fuzzy, or non-linear relationships. For our new TQM approach which focuses much on software quality control, System Dynamics could be a great tool.

This paper will discuss the possibilities of System Dynamics as a new tool for the TQM.

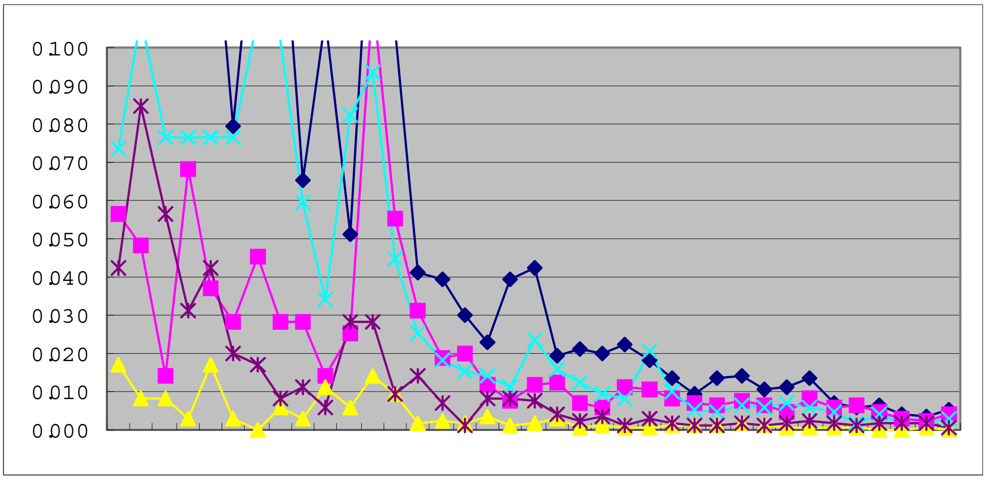
3. A Case Study – Software Quality Control and System Dynamics

3.1 Story

In a computer system development project where I have been working as a consultant since last year, users had been suffering from the poor quality of the system. They experience thousands of troubles every year and it takes long hours to fix them. In addition, they have to update the system every year within the limited budget. Therefore, only few moneys can be spent for the quality improvement activities. They have plan to expand their system to double in size in the next year, but they are wondering whether to do it.

3.2 Statistical Approach

At first, statistical approach was conducted to estimate the next year's situation. They collected the past trouble data and drew a chart to estimate the number of troubles in the future. The next graph shows the result.



This statistical analysis showed that troubles would disappear till the system expansion. However, users were not sure about this result because they did not understand the meaning of the curve fitted to the data. Some users feel that the quality of the system could be worse in the future. In addition, this graph says nothing about the effect of the future event, i.e., system expansion.

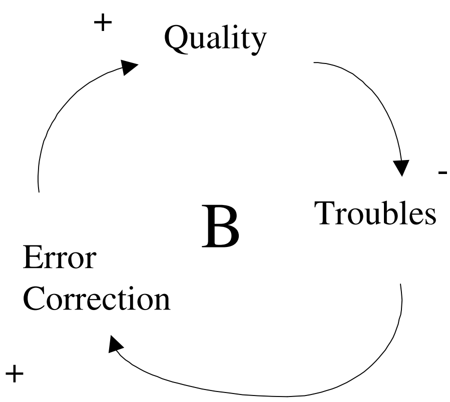
3.3 System Dynamics Approach

In the next step, I adopted the System Dynamics approach to solve this problem. I started to discuss and describe the current situation using the causal loop diagram. Users were also involved in this process to understand the structure of the project.

3.3.1 Current Situation Analysis

Quality Improvement Loop

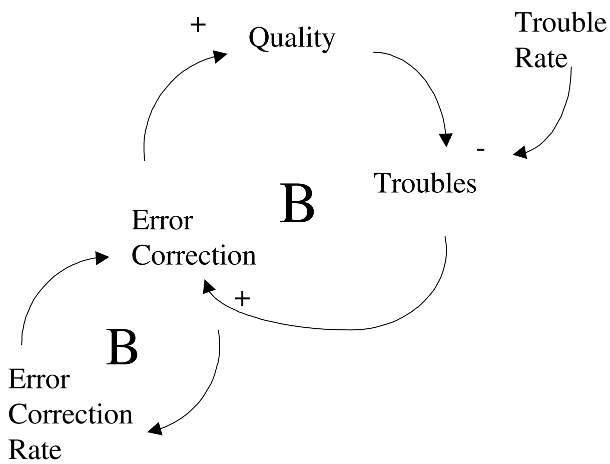
We started to understand the basic structure of the project. The main causal loop for the software quality is described as follows.



This diagram is read:

"If the quality level is lower, we suffer more troubles, which gives us more chances to correct errors, which improves the quality."

The main feedback loop of this quality management is the balancing loop, which reduces the number of the troubles along time. Here, the quality level is measured/estimated by the number of hidden bugs. As this first diagram is too simple to describe the reality, I added two more factors.

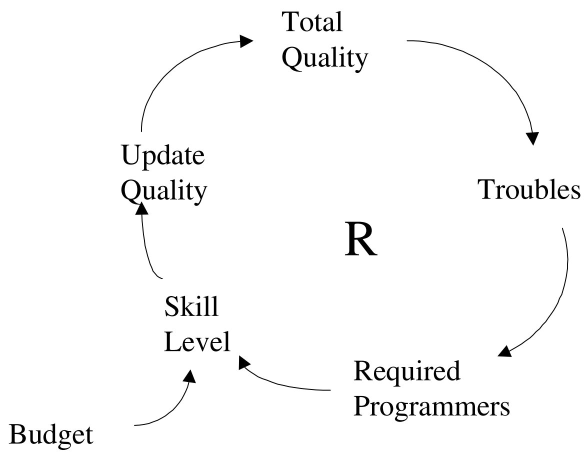


*Error correction rate: Not all errors are corrected because of the budget limit.
*Trouble rate: Not all hidden bugs cause troubles.

Trouble rate may be affected by the number of users or clients of the system. In this revised diagram, it takes more time to balance because of the second balancing loop.

Reinforcing Loop

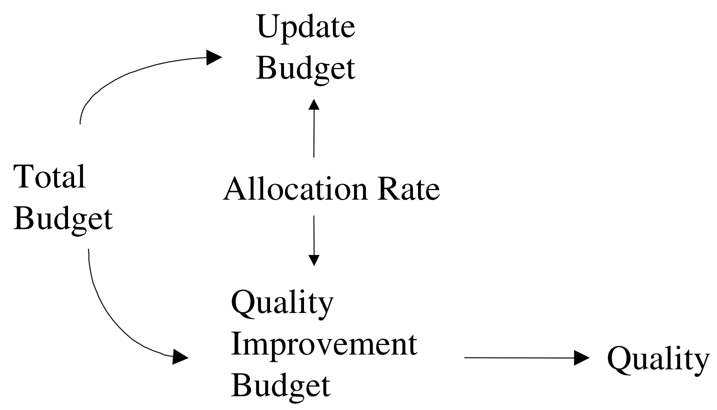
Users of this system feel that there are another factors that worsen the quality. Programmers' skill level greatly affects the quality of the newly developed or updated software, which then affects the total quality.



"If the quality level is lower, they suffer more troubles, which requires software makers more programmers. As the budget is limited, makers cannot allocate highly skilled programmers, which leads to the poor skill level of the project. It produces the poor quality software, which worsens the total software quality."

Budget allocation

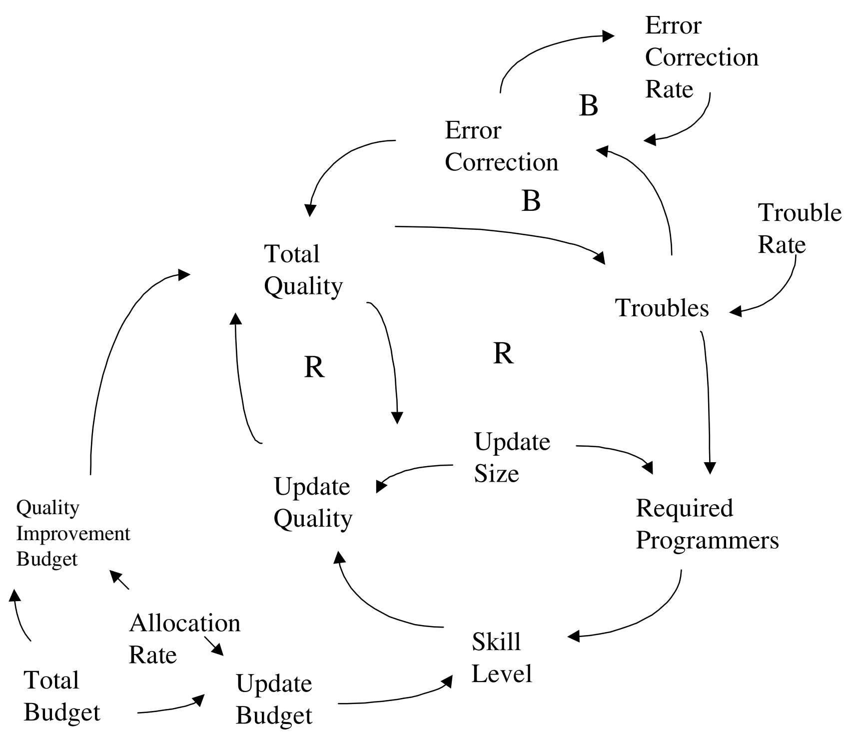
The budget allocation is described as follows.



In this project, most of the budget is allocated to update of the system and only few can be spent to improve the quality.

Total model

Below is the combination of the above diagrams.

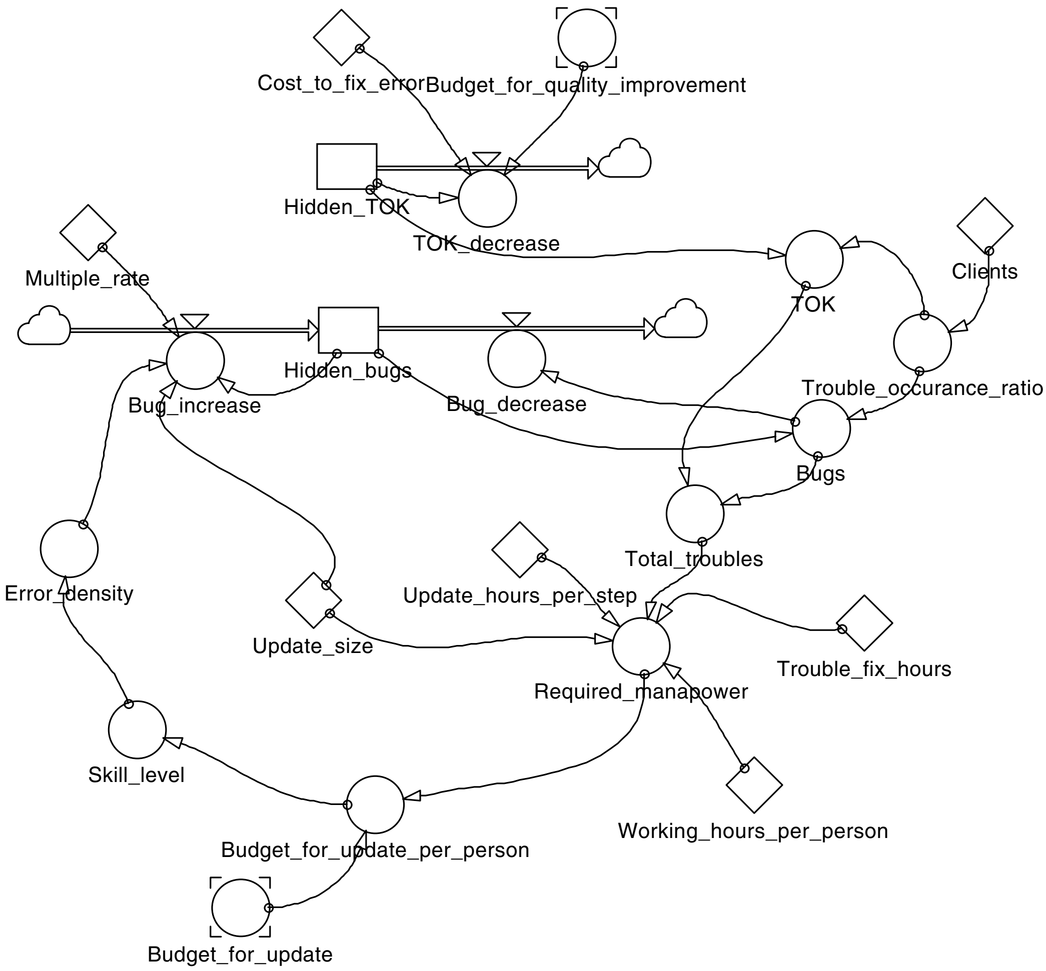


This step by step approach could make users of the system confident about the analysis.

Some people may feel this model is too simple to describe the complicated reality. However, this simple chart is enough to analyze the macro trend and see a big picture.

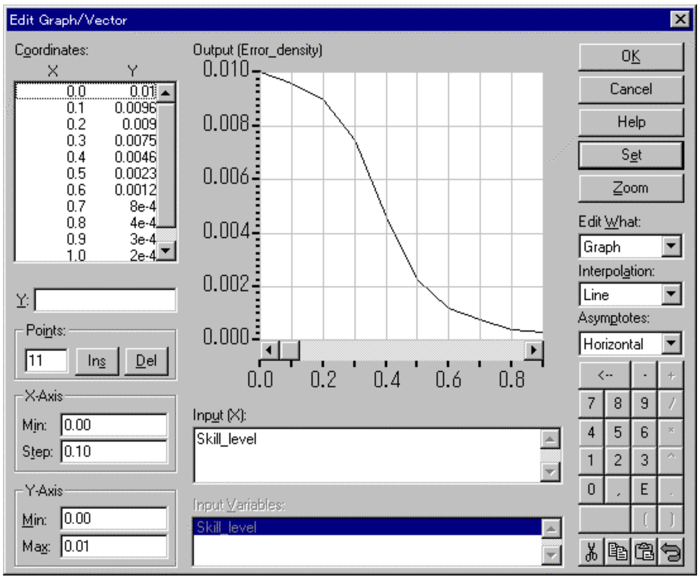
Stock and Flow Diagram

This model is already complicated enough to estimate the simulation results by our heads, which requires the help of computer technology. In order to use computer simulation, we drew the Stock and Flow Diagram.



Stock and Flow Diagram

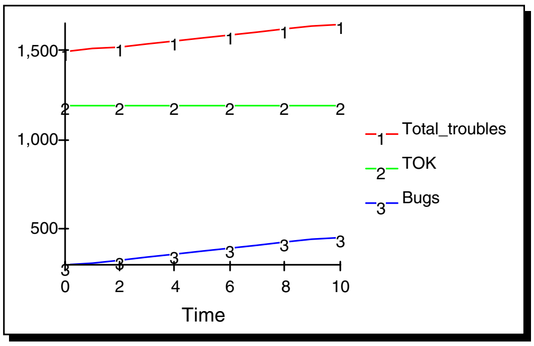
In this model, we used some graphs to describe the relationships without past data and equations. They are made from discussions between specialists. The graph below shows the relationship between the skill level and the quality. The error density of software is lower with highly skilled programmers.

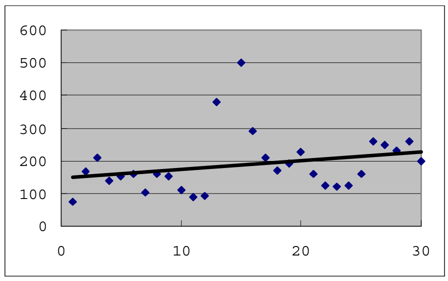


Some advanced SD tools, such as i-think and Powersim, have the function to input fuzzy relationships by dragging a mouse.

Simulation Result

Below left graph shows the result of the simulation. This model's validity is checked to the reality graph in the right.

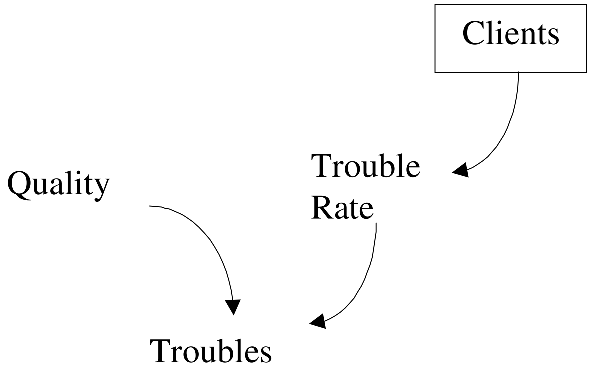




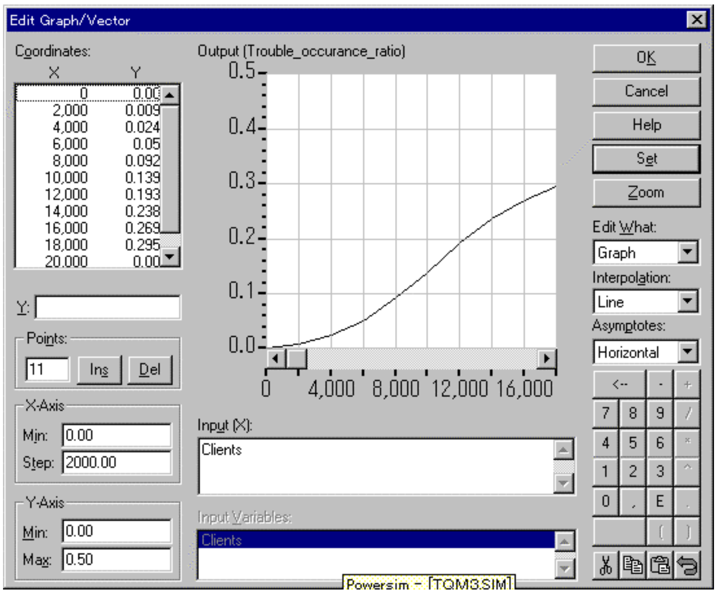
The number of troubles will gradually increase in the future. Users are more confident about this result than before as they participated in the cause-and-effect analysis process.

3.3.2 Future Estimation

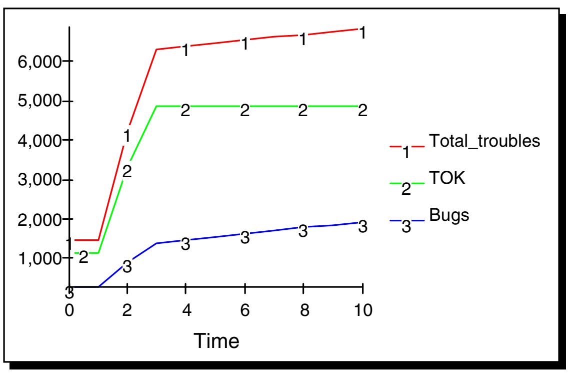
In order to estimate the effect of the expansion of the system, the model was revised. In the new model, Trouble Rate is affected by the number of users/clients. In the year 2, the number of users will increase from 6,000 to 10,000 (12,500 in the year 3).



The relationship between the Trouble Rate and the number of users/clients was described as the graph function below based on the discussion among the specialists.



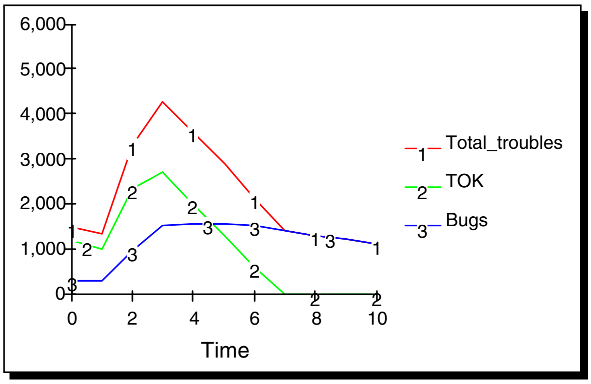
The result of the simulation is described in the next graph.



From this simulation, we realized that there is a possibility to suffer more than 6,000 troubles a year, which cannot be handled by the project staff. They need alternative plan for the system expansion. Thus, SD tool makes it possible to estimate the effect of the future event without past data.

3.3.3 Solution Selection

The start point to avoid the future disaster would be 'quality improvement'. If we allocate 10% of the budget to quality improvement, it will change the situation dramatically. (See the graph below.)



Even in this revised model, however, there will still be too many troubles. They need to introduce more effective quality improvement program before the system expansion. SD tool makes it easy to change models and simulate again.

4. Conclusion

As we saw, System Dynamics can be used in several phases of the QC story. System Dynamics can be a quite useful tool for TQM, especially for software quality control. Implications from this case study are:

4.1 Persuasive Tool

System Dynamics approach is more persuasive than the traditional statistical approach. It uses visual diagrams to describe the world, which helps us understand the cause and effect relationships. In addition, when users of the model are involved in the modeling process, it is more effective.

4.2 Manager's Laboratory

System Dynamics is a powerful tool because it provides a laboratory for managers. They can test their strategies before they implement them. In TQM activities, this greatly helps to select the best solution or plan.

4.3 Treating Fuzzy Relationships

In software quality control, there are more human factors that affect the quality than in hardware quality control. System Dynamics approach prepares some tools to treat these fuzzy relationships.

4.4 Combination with the Statistical Analysis

Statistical analysis cannot be denied with this new approach. It provides basis data for modeling. The right combination of the two approaches would make the best tool for the TQM.

References

Hidaka, S. (1998). Simulation Modeling: a New Tool for TQM, *Japan Quality Control Society Conference '98 Proceedings*

Hidaka, S. (1998). Simulation Modeling: a New Tool for FM, *World Workplace Europe '98 Proceedings*

Imai, M. (1986). *Kaizen*, McGraw-Hill

Oakland, J.S. (1993). *Total Quality Management: The route to improving performance (2Rev.ed)*, Butterworth Heinemann

Senge, P.M. (1990). *The Fifth Discipline: The art and practice of the learning organization*, Century Business