

# NSE Project Management Paradigm: Seamlessly Combined with the Project Development Process

Jay Xiong  
NSEsoftware, LLC., USA  
[jay@nsesoftware.com](mailto:jay@nsesoftware.com)  
[www.nsesoftware.com](http://www.nsesoftware.com)

## Abstract

This paper introduces NSE project management paradigm with which software development process and project management process are combined together seamlessly, the documents for project management are traceable with the implementation of the requirements and the source code to help the project management team and the project development teams find possible problems quickly and solve the problems in time.

## 1. The old-established software project management paradigm is outdated

Software projects need to be managed according to the project development plan, budget, and functions. Usually software project management tasks include:

- (1) project planning/scheduling
- (2) project monitoring
- (3) risk management
- (4) project cost estimation
- (5) process management
- (6) project documentation
- (7) unexpected event handling
- (8) people/team management

Since the term *software engineering* first appeared in the 1968 NATO Software Engineering Conference it has been more than 40 years past. Many books on software project management are published. But unfortunately, the project success rate is still very low – only about 30%. Why?

The root reason is that not only the old-established software engineering paradigm based on reductionism and superposition principle is outdated, the old-established software project management paradigm is also outdated because

- (1) It focuses on process rather than people as the first-order-effect on software development, violating John N. Warfield's "**Twenty laws of complexity**" that "The human being, language, reasoning through relationships, and archival representations are universal priors to science (i.e., there can be no science without each of them)"[War98]. As pointed by Alistair Cockburn that "The fundamental characteristics of 'people' have a first-order effect on software development" [Coc99]. Even if there are some models claiming people is the first-order-effect on software development, they handle people having positive effects only to offer better working conditions and tool support for the software development team – they ignore the negative effects from people. But the fact is, almost all defects are introduced into a software product by people – the customers and the developers.
- (2) It does not seize the principal contradiction – in most software organizations 75% or more of effort and cost are spent in software maintenance, but often the management team does

not attach importance to software maintenance – why? They know it is critical issue, but they feel powerless.

- (3) It can not efficiently handle the issue of changeability.
- (4) There is a lack of support methods and tools.
- (5) With it software development process and project management process are separated.
- (6) The Project management is always half a beat behind – hard to find problem in time and hard to solve the problem in time.
- (7) With it the project success rate is still very low – only about 30%, not acceptable to any other industries.
- (8) The root cause for the issues in software project management is that the old-established software development paradigm and the old-established software project management paradigm are based on reductionism and superposition principle that the whole of a complex system is the sum of its components, so that with the old-established software project management paradigm almost all software management tasks are performed partially and locally.

## **2. Outline of the NSE Project Management Paradigm**

The revolutionary solution offered by NSE for software project management will be described in detail in this paper later. Here is the outline of the solution:

- (1) It focuses on people rather than process. NSE software project management paradigm treats people with two side impacts: the positive side and the Negative side that almost all defects are introduced into a software product by people - the customer and the developers, so that it forces the management team not only to offer better working conditions and support to the software development team, but also to provide many efficient methods and tools to prevent people to introduce defects into a software product.
- (2) It seizes the principal contradiction - software maintenance. It makes both the software development process and the software maintenance process be managed together and forces side-effect prevention in the implementation of requirement changes and code modifications to greatly reduce the effort and cost spent in software maintenance. It is performed with support of various automated and self-maintainable traceability.
- (3) It can handle the issue of changeability better by preventing the side-effects in the implementation of requirement changes to ensure the quality of a modified product.
- (4) With NSE there are many methods and tools provided to support software project management, such as the method and tool for making the project management documents traceable to the implementation of requirements and source code.
- (5) With it software development process and project management process are combined together.
- (6) The Project management documents such as the schedule chart, project progress reports, the cost reports are traceable with the implementation of requirements and source code, so that the project management team can find the problems early and solve the problems in time.
- (7) With NSE software engineering paradigm including the NSE project management paradigm, it is possible for NSE to help software organization double their productivity and double their project success rate, compared with the old one.
- (8) The foundation for establishing NSE software project management paradigm is complexity

science. NSE software project management paradigm complies with the essential principles of complexity science, particularly the Nonlinearity principle, the Holism principle, the Self-Adaptability principle, and the Self-Organizing principle, so that with NSE software project management paradigm almost all software project management tasks are performed holistically and globally, such as the cost estimation is done with decomposition result of an entire software product preliminarily designed.

- (9) Real time communication support: with NSE software project management paradigm a project web site and the corresponding BBS traceable to the implementation of requirements and source code is required for real-time communication support.

### 3. The foundation of NSE Project Management Paradigm

The foundation for establishing NSE software project management paradigm is complexity science. It is established through the application of FDS (Five-Dimension Synthesis Method) framework as shown in Fig. 1.

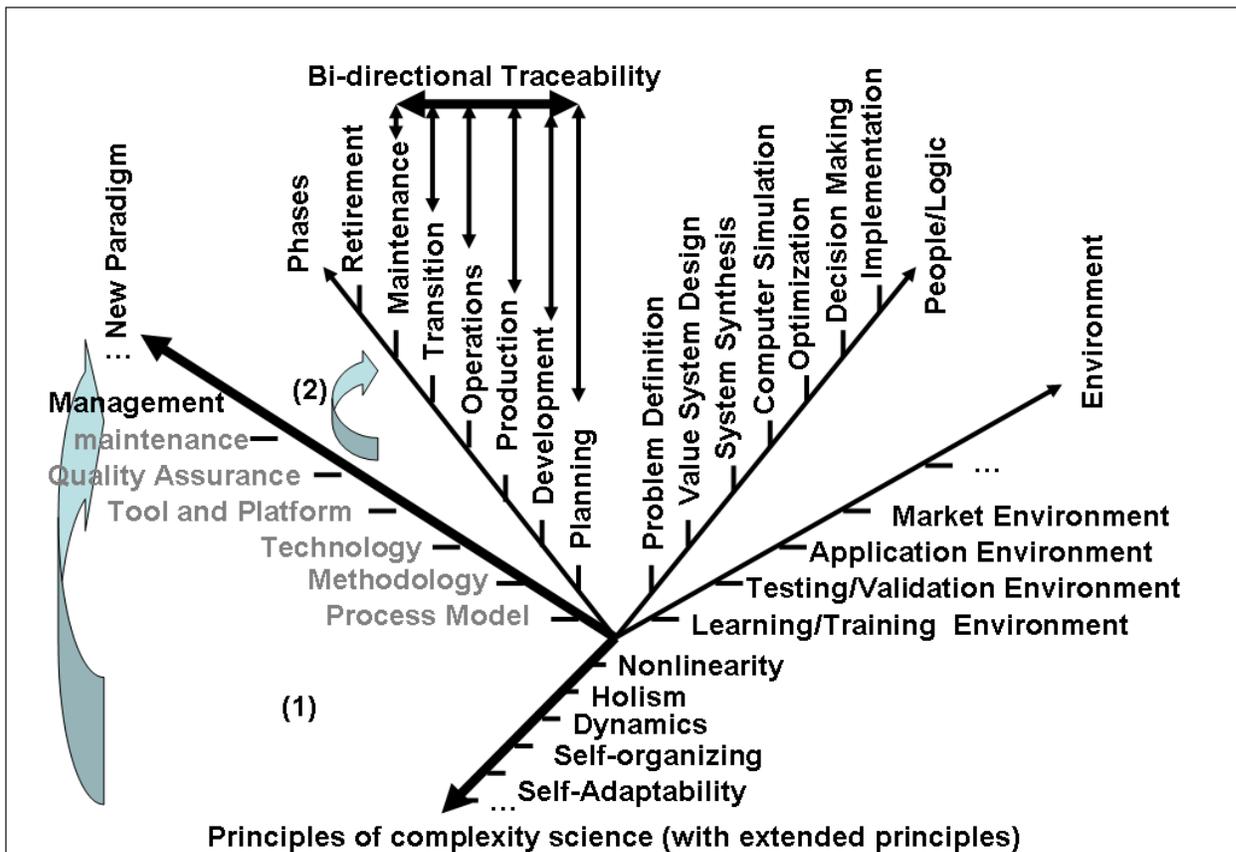


Fig. 1 The paradigm-shift framework, FDS (Five-Dimension Synthesis Method)

As shown in Fig. 1 NSE Project Management Paradigm complies with the essential principles of complexity science.

NSE Project Management Paradigm consists of project management strategy, methods, tools, and templates for project planning/scheduling, project monitoring, risk management, project cost estimation, process management, project documentation, unexpected event handling, people/team

management including training.

#### 4. The Strategy of NSE Project Management Paradigm

NSE Project Management Paradigm emphasizes on self-organization, self-adaptation, and self-maintenance.

Self-organization relies on feedback (positive and negative) , interaction, and balance of exploitation and exploration [Bon99], so that, for instance, with NSE Project Management Paradigm, all working product versions, even if it is a dummy whole system (as an embryo) , will be provided to customers for review to obtain customer's feedback.

Self-adaptation emphasizes on the support for software changeability and maintainability. With NSE requirement changes are welcome and implemented with side-effect prevention through various traceability. With NSE, a software product is maintainable not only in the produce development side, but also in the customer site – with NSE “Software” is redefined as and delivered to the customer with

- (1) instructions (computer programs) that when executed provide desired features, function, and performance;
- (2) data structures that enable the programs to adequately manipulate information; and
- (3) documents that describe the operation and use of the programs (including the test case script files too); **plus**
- (4) the database built though static and dynamic measurement of the programs; and**
- (5) a set of Assisted Online Agents (AOA, artificial intelligence tools working with the database) for supporting testability, reliability, visibility, changeability, conformity, and traceability to make the software program maintainable, adaptive.**

#### 5. People-Oriented

With NSE People-oriented management emphasizes on

- (1) innovation and continuous improvement in the existing products and services to match the fast changing demands of the market;
- (2) establishing a set-up with an environment to enhance operational efficiency of the organization;
- (3) developing human resources, by taking care of the needs and aspirations relating to career progression and job satisfaction through involvement, participation, training, and commitment [Kha02];
- (4) understanding of that people is nonlinear, easy to make mistakes in reading, writing, thinking, making decisions, communication, etc., - almost all defects are introduced into a software product by people (customers and developers), so that with NSE a set of methods and tools are developed to prevent the defects introduced into a software product by people. For instance, we know the obtained function decomposition result using Use Case is not traceable and not directly executable, then how do we know where the defect exist? With NSE, HAETVE (Holistic, Actor-Action and Event-Response driven, Traceable, Visual, and Executable) requirement development technique and the tool are applied to prevent defects introduced into a software by people through traceability for static review and program execution for dynamic testing – see Fig 2 and Fig.3.

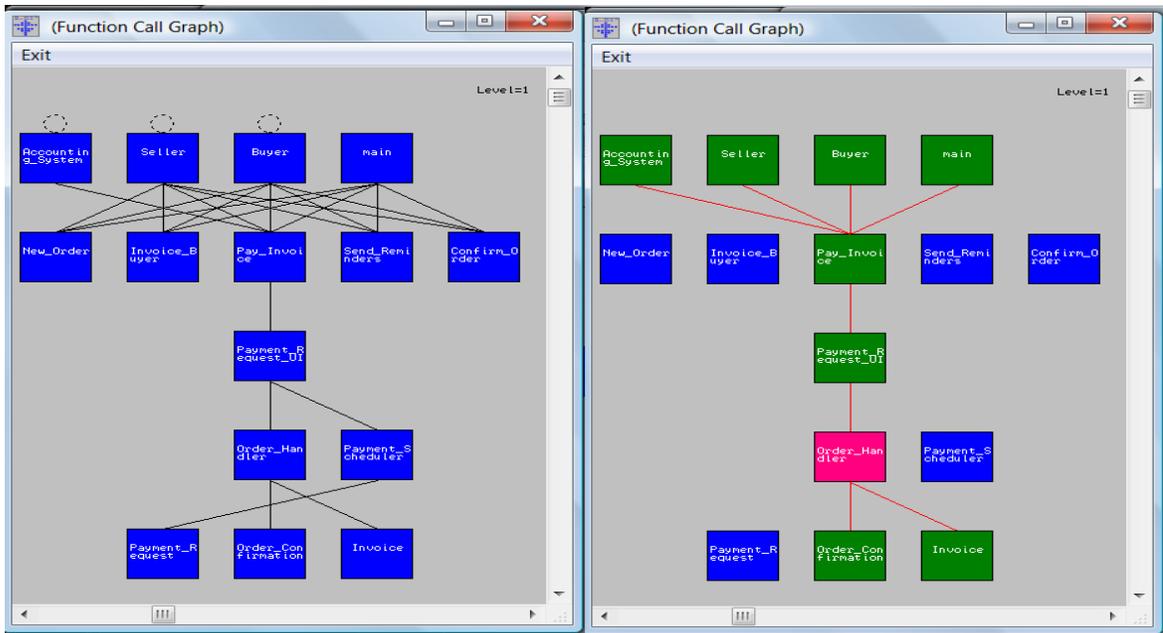


Fig. 2 The requirement development result using HAETVE technique is traceable for defect finding and removing

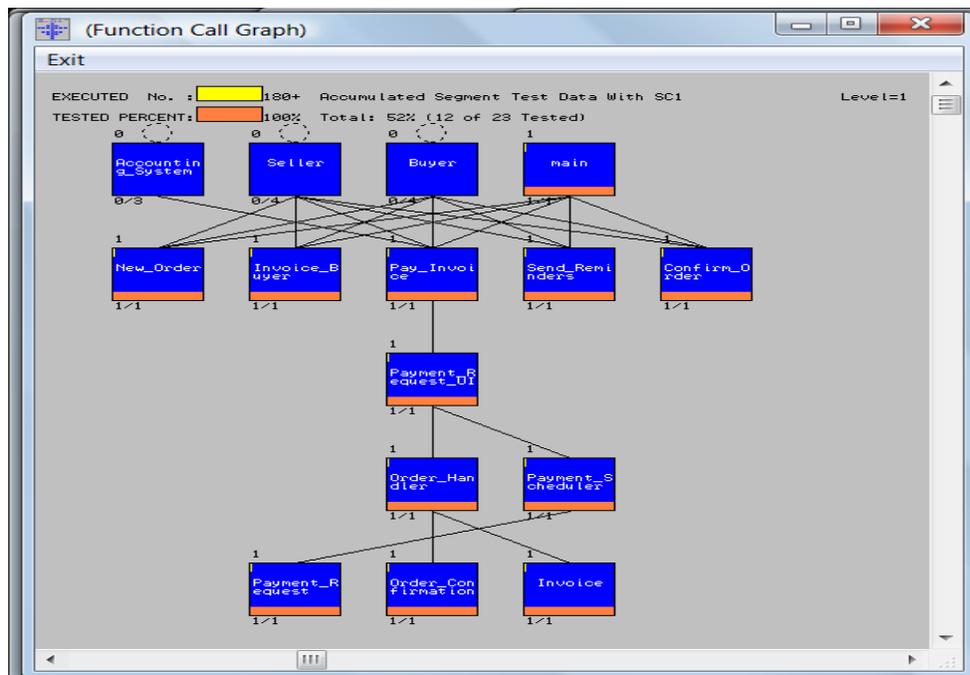


Fig. 3 The requirement development result using HAETVE is executable for defect removing (as shown in this figure, all “Actions” are executed except the “Actors”)

## 6. Focusing on maintenance

With the old-established software engineering paradigm based on reductionism and superposition principle, linear process modules are used, making defects easy to propagate from upstream to downstream, software maintenance is performed partially and locally, so that 75% or more

of the total efforts and total cost are spent in software maintenance. It is clear that for being able to double software productivity and halve software development cost, we must solve the issues with software maintenance.

With NSE the solution is simple:

- (1) Combining software development process and maintenance process together closely, supporting requirement change at any stage through side-effect prevention.
- (2) Greatly reducing the defects introduced into a software product and the defects propagated into software maintenance through defect prevention and defect propagation prevention.
- (3) Greatly reducing the new defects introduced into a software in the maintenance phase by performing the implementation of requirement changes and code modification holistically and globally with side-effect prevention supported by various traceability.

## 7. More Method and Tool support

Almost all required methods and tools for supporting software project management are developed and provided, particularly

- (1) the method and tool for making project management documents (such as the schedule charts and cost reports) traceable with the implementation of requirements and the source code.
- (2) the methods and tools of NSE software visualization paradigm to make the entire software development process and management process visible, and the work products visible.
- (3) The methods and tools for cost estimation using call graph shown in J-Chart notations, see Fig. 4.

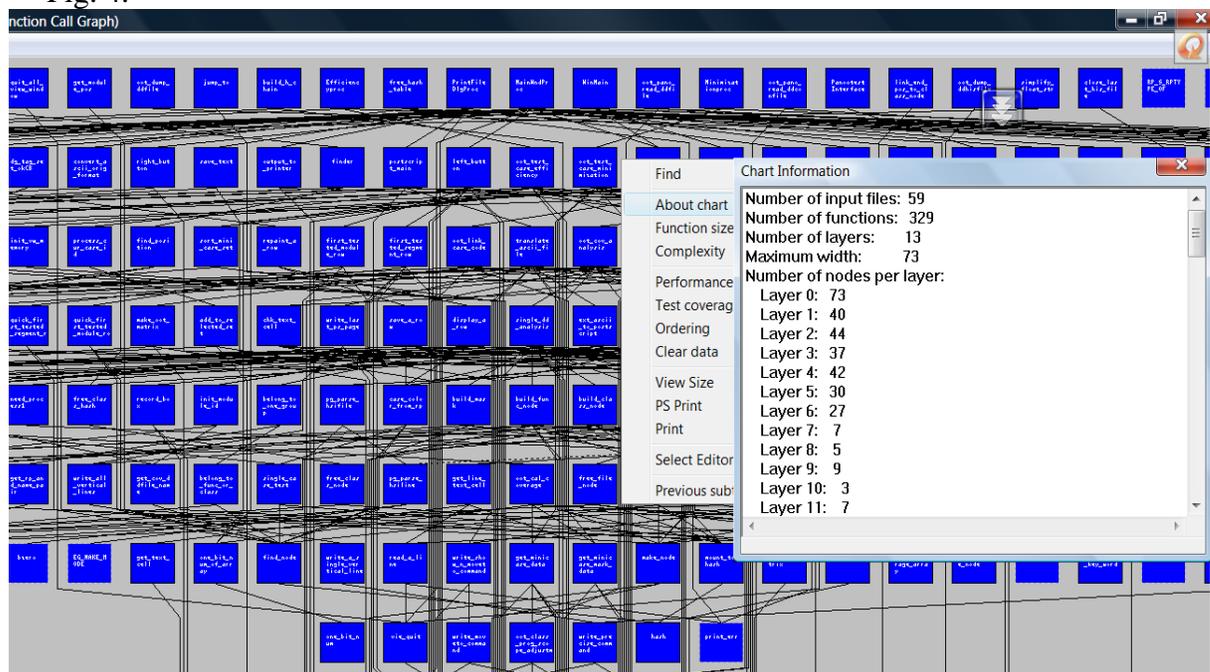


Fig. 4 A system call graph shown in J-Chart for cost estimation

- (4) Precise productivity measurement methods and tools – see an application example shown in Fig. 5.

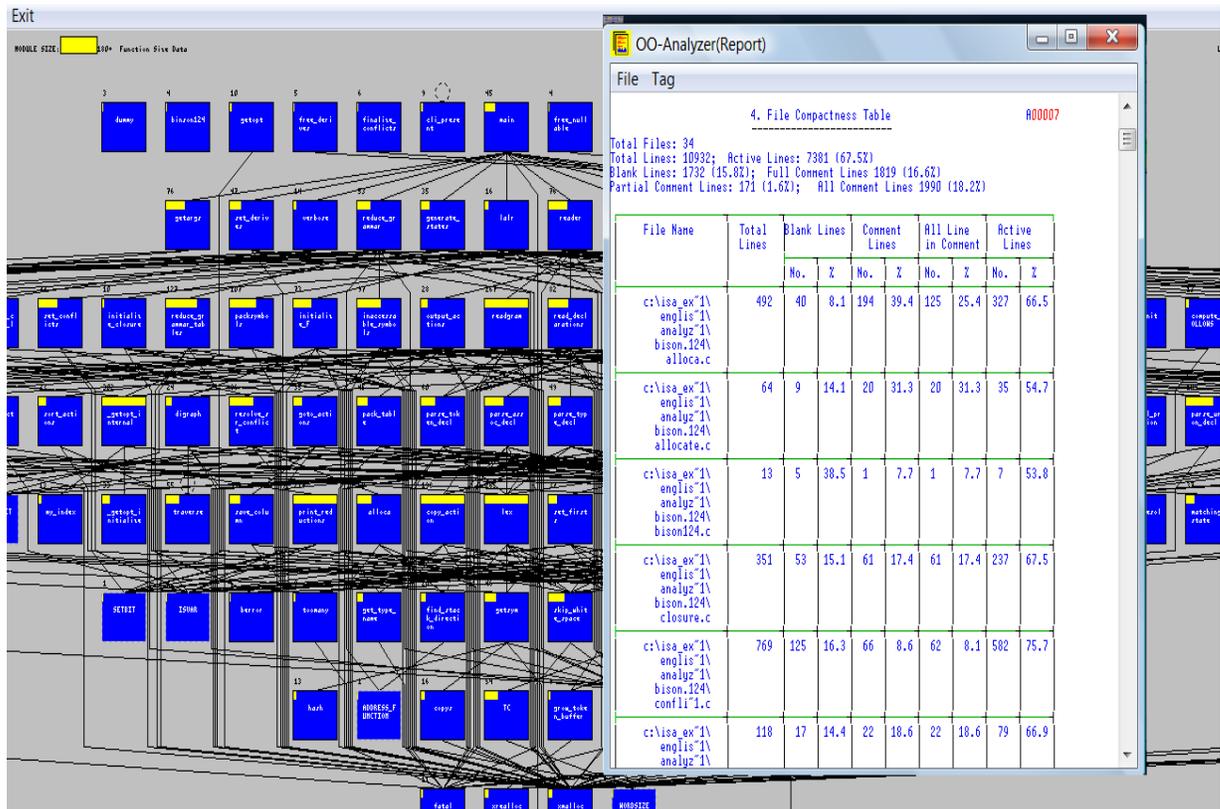


Fig. 5 Precise productivity measurement support

## 8. Combination of Produce Development and Project Management together

One of the root causes of software failures is that the project management process is separated from the product development process. In the article “Social and Technical Reasons for Software Project Failures” Capers Jones pointed there are five root causes for software failures:

1. Root causes of inaccurate estimating and schedule planning.
2. Root causes of incorrect and optimistic status reporting.
3. Root causes of unrealistic schedule pressures.
4. Root causes of new and changing requirements during development.
5. Root causes of inadequate quality control. [Jon06]

I think the fundamental root cause for software project failures is that the old-established software engineering paradigm is based on reductionism and superposition principle, so that with it almost all software engineering tasks and project management tasks are performed partially and locally. But the root causes Capers Jones pointed out are also existing with today’s software development.

How can we solve those issues? First, those problems should be handled holistically and globally; the project management process and the product development process should be combined together to make the project management documents such as the schedule and cost reports traceable with the requirement implementation and the source code.

“So it is today. Schedule disaster, functional misfits, and system bugs all arise because the left hand doesn’t know what the right hand is doing.”[Bro95-P74] – by combining the project management process and the product development process together, making the work products of project management and the work products of the product development traceable, and setting project/product web site and BBS for real time communication, will **make the left hand knows what the right hand is doing, and the right hand knows what the left hand is doing** to solve those issues efficiently. A schedule chart traced and opened when performing forward tracing from a requirement/test case is shown in Fig. 6.

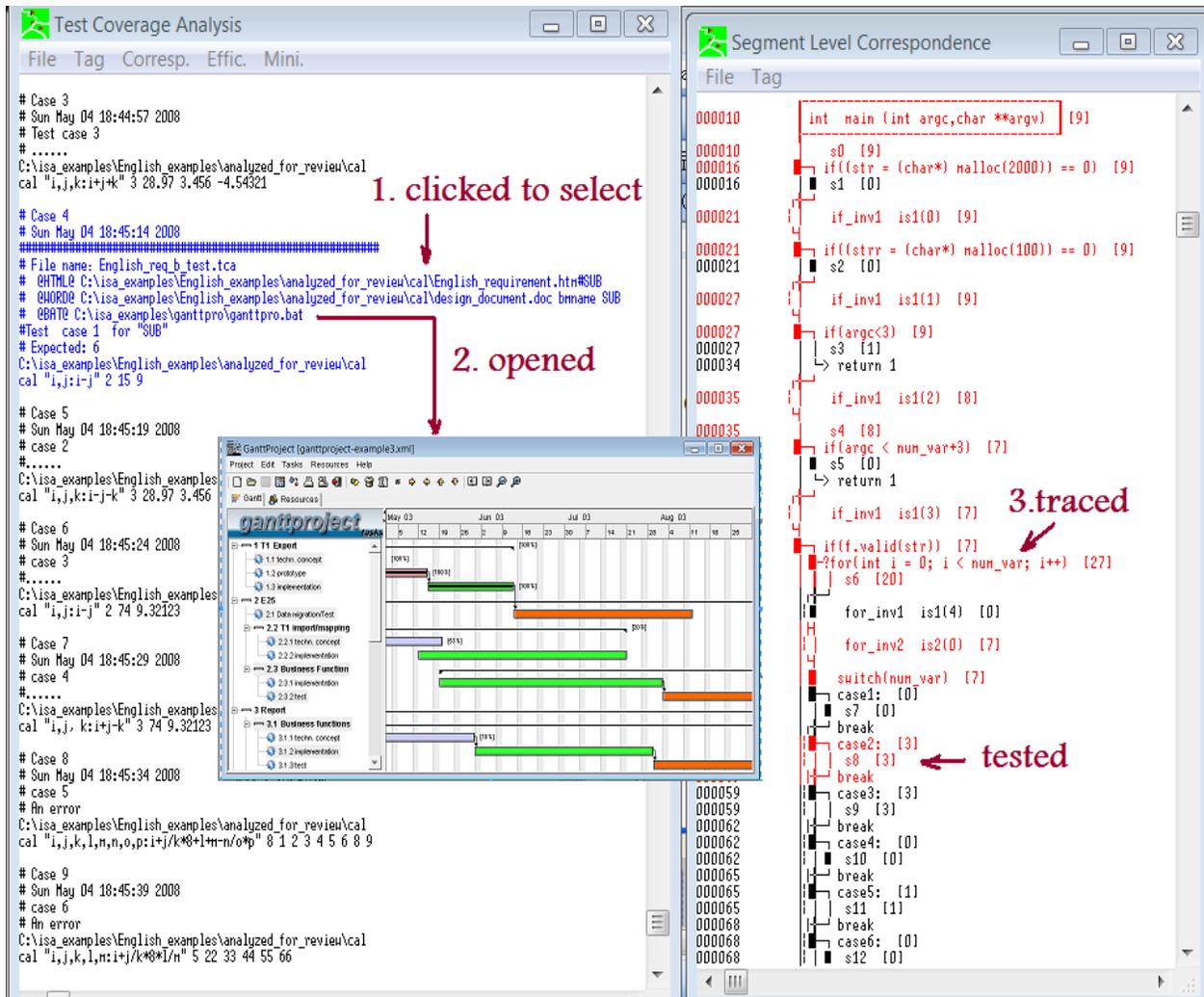


Fig. 6 A schedule chart traced and opened when performing forward tracing for q requirement/test case

A sample web page traced and opened when performing forward tracing from a requirement/test case is shown in Fig. 7.

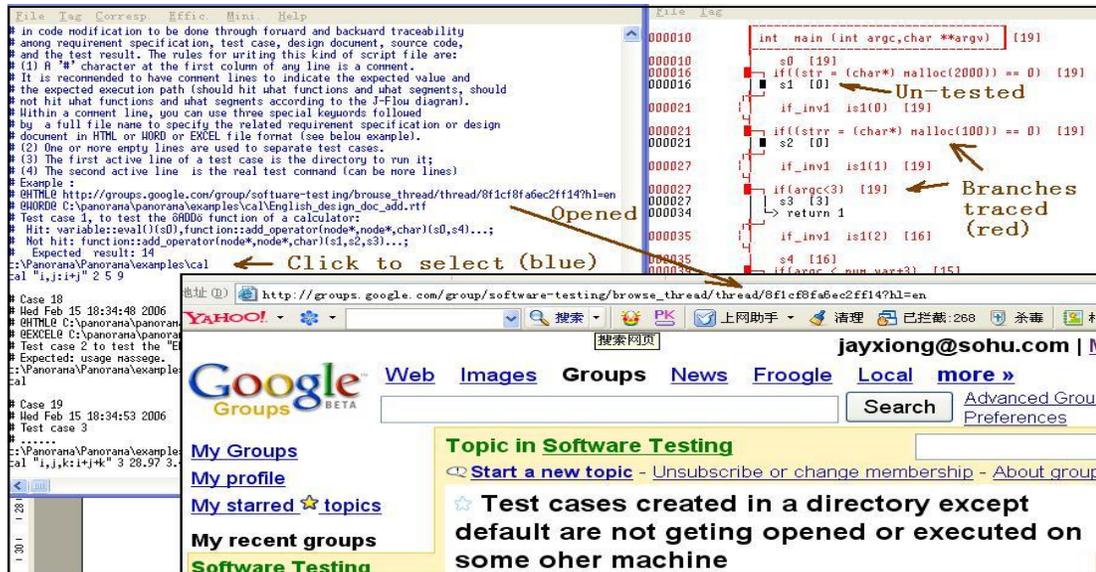


Fig. 7 An application example of tracing a requirement/test case to open a related web page

## 9. Finding Problems Early and Solving the Problems in Time

As pointed by Frederick P. Jr. Brooks, “When one hears of disastrous schedule slippage in a project, he imagines that a series of major calamities must have befallen it. Usually, however, the disaster is due to termites, not tornadoes; and the schedule has slipped imperceptibly but inexorably. Indeed, major calamities are easier to handle; one responds with major force, radical reorganization, the invention of new approaches. The whole team rises to the occasion.”, “But the day-by-day slippage is harder to recognize, harder to present, harder to make up.” [Bro95-P154].

The benefits by combining software development process and software project management process together, and making the work products of software development and the work products of project management traceable, will be able to find problems early, and solve the problems in time.

## 10. Quality Management

As pointed by Roger S. Pressman, “Software engineering will change – of that we can be certain. But regardless of how radical the changes are, we can be assured that quality will never loss its importance and that effective analysis and design and competent testing will always have a place in the development of computer based systems.” [Pre95-P867].

With NSE software quality assurance and quality management is performed from the first step down to the final step – the retirement of a software product.

## 11. Multiple-Project management

NSE software project management paradigm supports multiple-project development and management – two or more related projects’ documents, including the management documents and the progress reports, can be traced to each other as shown in Fig. 8. With the traceability, Events, progress, and issues in one project can be viewed by the management team in another project to take

corresponding actions and help each to other.

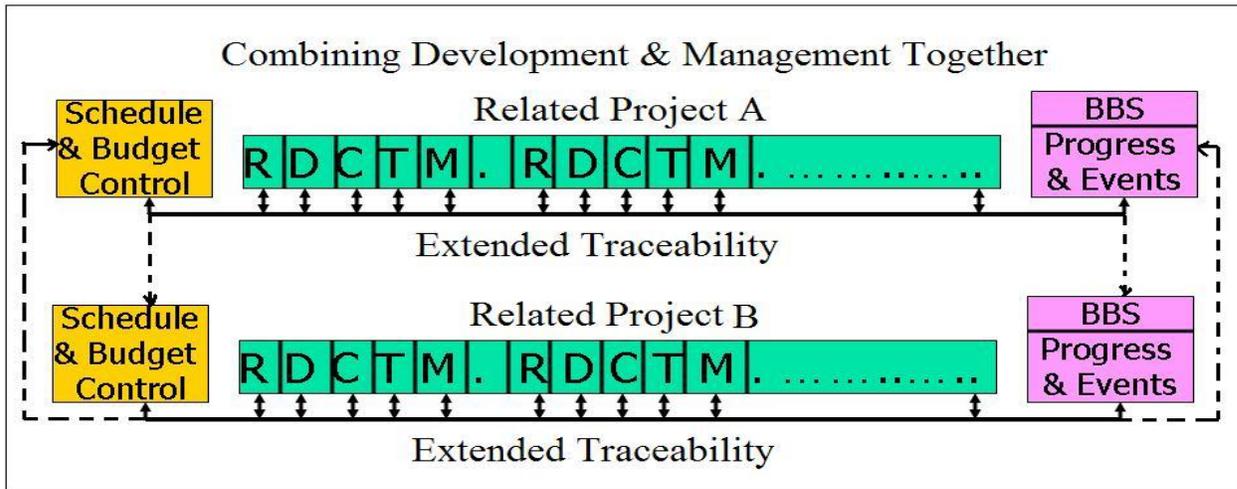


Fig. 8 Multiple Project Management

## 12. Conclusion

Software project management is an important factor for project success. The old-established software project management paradigm is outdated which is based on reductionism and superposition principle so that with it almost all project management tasks are performed partially and locally, such as the management for software changes.

NSE software project management paradigm is based on complexity science by complying with the essential principles of complexity science, particularly the Nonlinearity principle, the Holism principle, the Self-Organization principle, and the Self-Adaptation principle, so that with NSE software project management paradigm almost all software project management tasks are performed holistically and globally.

The most important feature of NSE project management paradigm is that software project management process is combined with software development process – the management materials such as the schedule charts, cost reports, progress reports, unexpected events reports are traceable with the implementation of requirements and the source code, so that the management team can find possible problems early and solve the problems in time.

People-oriented and maintenance focusing are also the important features of NSE project management paradigm for efficiently increasing software project success rate.

Project web pages and BBS title pages are traceable with the implementation of requirements and the source code – it is also an important feature of NSE.

## References

- [Bon99] Eric. Bonabeau, Marco Dorigo, and Guy Theraulaz (1999). *Swarm intelligence: from natural to artificial systems*. pp.9-11.
- [Bro95-p74] Brooks, Frederick P. Jr., “The Mythical Man-Month”, Addison Wesley, 1995, P74
- [Bro95-P154] Brooks, Frederick P. Jr., “The Mythical Man-Month”, Addison Wesley, 1995, P154.

- [Coc99] Alistair A.R. Cockburn, *Characterizing people as non-linear, first-order components in software development*, Humans and Technology, HaT Technical Report 1999.03, Oct 21, 1999
- [Jon06] Jones, Capers, Social and Technical Reasons for Software Project Failures, **CrossTalk**, Jun 2006 Issue
- [Kha02] Akram Khatoon, *People-oriented management*, DAWN , November , 2002,  
<http://www.dawn.com/2002/11/11/ebr19.htm>
- [Pre05-p867] Pressman, Roger S., “Software Engineering: A Practitioner’s Approach”, McGraw-Hill, 2005, P867
- [War98] John N. Warfield, *Twenty laws of complexity: science applicable in organizations*, Wiley InterScience, ([www.interscience.wiley.com](http://www.interscience.wiley.com)), 1998,  
<http://www3.interscience.wiley.com/journal/71007260/abstract?CRETRY=1&SRETRY=0>