Quality Driven Software Development

Ákos Szőke

In today's world of management of software engineering, we come across a feature called imprecision, which is associated with the following main characteristics of the software development process: costs, schedules, and quality. Now, with state-of-the-art technology we are able to provide plentiful functionality, and customers demand it at high quality. The narrow definition of the quality is "conforms to requirements and is fit to use". From the customer's view, satisfaction after the delivering of the product is the ultimate validation of the product quality. From the producer's perspective, developing and producing the product in accordance with the specifications is the path to achieving quality.

Unified Modeling Language (UML) is capable of modeling software product. Every product is made of less or more units which realize different part of the problem that we should solve. Software Process Engineering Metamodel (SPEM [1]) is an UML compatible metamodel, so UML is capable of modeling software development process itself too. Therefore, we can describe both software development process and product to serve input parameters of the well-known COCOMO II [2] cost estimator which can help us to estimate the cost of developing software product using product and development project cost factors.

The concept of defect removal effectiveness and its measurement are in the centre of software development. Increasing defect removal effectiveness can conduct to product quality improvement and reductions in development time. Since defect removal and its efficiency is one of the top expenses in any software project and it greatly affects schedules, static validation techniques and quality management models are worth considering.

The UML provided visual programming and automatic code generation are capable of eliminating numerous design and programming defects. But this approach can not provide semantic correction of the specification, which is the one of the major factor of defects. The more defects are reveled later, the more expensive and time-consuming of the correction. With the help of static validation of the specification (without running of the program which developed according to the specification) we are able to check the completeness of the specification which is written in UML notation.

Other important methods for improving quality are Quality Management Models which monitor and manage the quality of the software while it is under development, therefore these models can provide early signs of warning or improvement so that timely actions can be planned and implemented in due term [3].

As contest in the software industry became sharp, the importance of productivity and quality in software development have started to increase more and more. So we should complement the classical "what" and "how" questions of design decision with "why" and "how much" questions to find the optimal cost, schedule, human resource allocation and quality values according to the customer's and the producer's aspect.

References

[1] OMG, Software Process Engineering Metamodel Specification, OMG 2002

[2] Center for Software Engineering, Constructive Cost Model (COCOMO) URL: `http://sunset.usc.edu`

[3] S. H. Kan, Metrics and Models in Software Quality Engineering, Addison Wesley 2002

[4] M. L. Hutcheson, Software Testing Fundamentals: Methods and Metrics, John Wiley and Sons 2003

[5] N. E. Fenton, S. L. Pfleeger, Software Metrics 2nd Edition, PWS Publishing Company 1997

[6] M. Paulk, B. Curtis, M. Chrissis, C. Weber, Capability Maturity Model for Software, Tech. Report 1993 CMU/SEI