

Position paper for WESS:

Session 2. What are the differences between maintenance skills and those of development?

(Session chair: Anneliese von Mayrhauser, Colorado State University)

Just-in-time-comprehension vs. the Full-coverage strategy.

Janice Singer  
Timothy Lethbridge

Under the auspices of the Knowledge Based Reverse Engineering Group, we have longitudinally followed one software engineer (SE), who we will call X, from the time he entered a large telecommunications company two years ago until today. During this period we met with him at one to six week intervals. During the meetings, X told us what did since the last visit; i.e., what tools he used, who he consulted, what problems arose (either organizational, knowledge based, or technical), etc. We also asked X at most meetings to draw a "map" of his current conception of the system, and observed him working for half an hour.

Our data indicate that X uses two different, yet complementary, strategies when writing source code. While doing routine maintenance and bug fixes, X employed a Just-in-time Comprehension strategy. Here X attempted to comprehend only that portion of the code that was necessary for solving the current problem. X did not work towards building a global conceptual model of every aspect of the system, but rather he iteratively built a partial model of the system as he solved the problem. This partial model tended to be forgotten as X moved on to other problems. Using JITC, X tended to concentrate on the code, and on running simulations in the laboratory (this is not to say he didn't use other information, but these were his primary sources).

While doing development work (adding a new feature to the system), X employed a Full-coverage strategy. Here X spent much more time trying to understand how all features of the system work together. He needed to ensure that his conceptual model of the architecture was appropriate so that the new feature addition would succeed (i.e., all interfaces work correctly and no feature interactions occur). To achieve this, he believed that he had to understand overall how the system worked. Here, X looked at the source, consulted documentation, people, ancillary texts, and additionally, wrote functional analysis and specification documents.

The JITC and Full-coverage strategies have important similarities to Littman, Pinto, Letovsky, and Soloway's As-needed and Systematic strategies. We have extended their results by showing that these strategies are applied not only when SEs are looking at very small pieces of unfamiliar code, but also when they are involved in solving problems in huge and somewhat familiar systems. Additionally, we have shown that the same SE can use both strategies. What determines usage is not 'SE-type,' but rather SE goal. Because of this, we have chosen to change the names Littman, et al. coined. We do see one strategy as being 'better' for program comprehension. Rather, we believe that the two different strategies help software engineers achieve different goals.

When working with huge programs with millions of lines of code under time pressure to perform changes, it makes little sense to try to understand the source code of the entire system. Thus for routine maintenance, SEs will employ the path of least resistance and use JITC. However, when development calls for understanding how a new feature can be fitted into an existing system, Full-coverage will be the strategy of choice. We hypothesize that this difference in strategy usage is dictated by the type of work that SEs are doing because development and maintenance are not simply two sides of the same coin, but rather qualitatively different activities that require different approaches.

#### Descriptions of maintenance background and experience:

Timothy C. Lethbridge has performed software maintenance in both private and public sector organizations for over 15 years. As an assistant professor at the University of Ottawa, he now focuses on studying how industrial software engineers perform development and maintenance, as well as what tools are most useful for them in these activities.