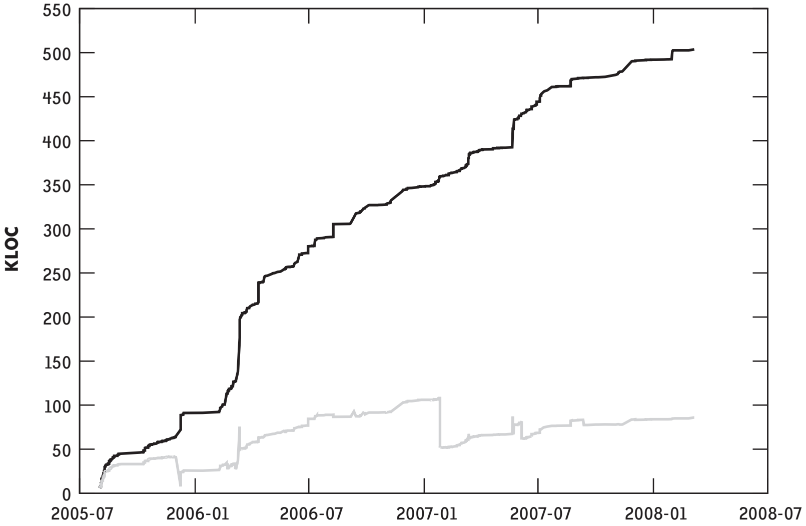

FIGURE 9-13. Accumulated code in JPC

A continuous cycle of purge, rewrite, and refine is difficult to achieve, although it is easier in an academic environment. Provided you don't get too emotionally attached to your code, the process of code deletion can be cathartic, not only for the codebase, but also for the developer's attitude toward it. Henry Ford was right: almost invariably your second attempt will be better. As long as this series of iterative improvements has a termination condition, all will be well.

So, the road to a "Beautiful Architecture" would be a four-step plan:

1. Take a large and complex problem in its entirety, and find a set of credible simpler stages that would enable a complete end-to-end prototype system to be built. Each stage represents a system simpler and less capable than the final goal, but each stage can be tested within its design limitations as part of a complete system prototype rather than as individual prototypes of small parts of the final design (i.e., a more traditional unit test).
2. Before building each part of each stage, be clear what aspect is being developed and why. Ideally, the bottlenecks in each stage can be easily identified and improvements to these will be the main targets for the next stage or stages. Look hard to find ways to prove the methodology in principle before embarking on large chunks of work—even for each part of each stage.