Haste Makes Waste When You Over-Staff to Achieve Schedule Compression

Doug Putnam  
QSM Vice President

The Research

Schedule pressures often force development managers to seek strategies to compress project time scales. Typical strategies might include adding more people, investing in development tools, improving hardware, and improving development methods. Is it possible to see which of these strategies is most effective? In this research we concentrate on the application of people to achieve time compression.

Our Method

Managers often resort to the people strategy because it's the most straightforward to implement. People can be easily redeployed and no major capital investment is required. But how much schedule compression is achieved by adding people? And are there any side effects?

QSM maintains a metrics database of over 4,000 completed, high-confidence projects. In this study we analyzed 3 separate data sets:

- Information systems completed after 1994
- Engineering systems completed after 1994
- Real-time embedded systems completed after 1994

For each set we separated the projects into small teams (fewer than 5 people) and large teams (more than 20 people). We then graphed the data showing schedule, effort and defects versus the size of the system that was built. This shows how much schedule compression was achieved by the larger teams as well as any insights into the side effects of cost and defects.

The Results

To illustrate the findings we show the data from the information systems data set. Both the engineering and real-time embedded sets showed similar results - in fact, they were more extreme.

Peak Staffing

In this figure we show peak staffing plotted at the size of the developed source lines of code. The red squares are projects that used 5 or fewer people; the blue squares are projects that used 20 or more people. The line through the data shows the average staff at a given size. For comparison purposes the graph shows the average staffing at 100,000 source lines of code with team sizes of 4 and 32 people.
Development Effort

The second figure shows the effort/cost differences experienced with the different strategies. For a typical project of 100,000 source lines of code, a 32-person team on average would have used 178 person-months of effort ($2.1 million @ $12,000 per person-month). The 4-person team would have used 24.5 person-months ($294,000 @ $12,000 per person-month). The difference between the strategies is approximately $1.84 million.

Development Schedule

How much schedule compression was achieved? Not much! The difference between the large team and the small team approaches for the average project of 100,000 source lines of code is approximately 1 calendar week. We do note that there is more variability in the small team data set, but we will analyze this in a future piece on optimum team sizes.
Defect Creation

How does one explain the fact that all the additional manpower and cost had a very minimal impact on schedule? One clue is that the large teams created significantly more defects. For the average project of 100,000 source lines of code, the large teams created over 5 times as many defects. The increased volume of defects created more rework cycles, negating the schedule benefits of the additional people.

Risk Management Strategies

Although adding people to a project might seem like a straightforward remedy for schedule compression, our research shows that it is often counterproductive. Anyone who has driven a car knows that maximum acceleration burns fuel at a significantly higher rate and provides only marginal time compression in getting you to the final destination. Software projects seem to exhibit this same nonlinear behavior.

There may be business situations where weeks of time are critical in the launch of a new product or service. The additional
cost incurred by adopting the manpower-intensive approach may be trivial compared to the revenue or benefits derived by earlier delivery. If you chose the manpower intensive strategy, be aware of the risks, and condition your customers to have realistic expectations. Carefully select the additional people to complement the existing team. Also, be sure to plan additional testing resources to handle the increased volume of defects.