SPE Experience: An Economic Analysis

Position Paper: Mary Hesselgrave, How Many? How Fast?

The Business Point of View

Project managers believe in the necessity of choosing two out of three: schedule, cost, and quality. While performance engineering adds to the quality of a project, it is perceived to have a significant negative effect on the schedule and cost, primarily because of the required staff effort. However, a complete analysis of the value of SPE often shows that it has a net economic benefit for the business.

Business Value of Performance

A business value analysis begins with determining the business ***need*** for performance quality:

- **How many?** What is the workload arrival rate distribution?
- **How fast?** What is the value of a response time second? For example, does saved processing time save lives, reduce the number of employees required, or decrease the probability of web page abandonment?
- **How expensive?** What is an acceptable cost per workload unit at the required service level? What is a competitive price point?
- **What is the cost of failure?** A vendor or service provider may pay penalties for not meeting a Service Level Agreement.

SPE Costs

Costs of integrating SPE into a project process may include:

- Cost of a performance engineer's time.
    - One performance engineer may be able to work with a group of projects. For projects involving less than 500 to 1000 staff, part-time use of a performance engineer can be cost-effective, particularly when the same person works with related systems, and/or with the same system in different development cycles.
    - Total performance staff effort may follow a 1 to 75 ratio, rather than a 1 to 500 or 1 to 1000 ratio. This includes allocation of the time of non-specialists for specific performance-related tasks such as performance quality assurance.
    - The estimated cost for a software performance engineer is:
        - $200K loaded salary per year for SPE in organization
        - $2K per day for an internal SPE consultant.
        - $2.5K per day for an external SPE consultant.
- Cost of staff time to learn unfamiliar SPE process tasks, to interact with a performance engineer, and to carry out SPE process tasks, such as preparation for and participation in an architecture review, or budgeting, estimating, and measuring resource use throughout a system construction effort. A reasonable rule of thumb is to add 1% to 5% to the estimated effort for systems engineers, architects, developers, and testers. The time spent should be proportional the

business value of performance, and the risk of performance failure. The estimated cost of staff time is $150K loaded salary per year for a typical technical employee.

- Cost of extra hardware for performance testing. There is a large variation in this cost, from essentially nothing to millions of dollars. When performance is critical to project success, large-scale duplication of deployment configurations with input simulators can be costly. For many projects, automated performance tests can be run during off-hours on available test systems. In some circumstances, a focus on in-depth performance analysis can reduce the size of needed test configurations.
- Cost of performance-modeling software and hardware. There is a large range of software that can meet modeling needs, depending on the type of project. Spreadsheets and inexpensive (<$1000) modeling software can handle many modeling needs. More detailed modeling may be more efficient with the support of more powerful software, and is appropriate for projects with higher performance risks.

Costs of not integrating SPE into a project process may include

- Schedule delay for late performance fixes.
- Increased development costs because of tuning and rework for development and quality assurance.
- Increased software maintenance costs for handling performance problems and for working with software mangled by ad hoc performance fixes.
- Increased hardware costs due to inefficient software and lack of time to fix it.
- Projects cancelled because performance objectives cannot be met.
- Increased operational costs due to systems with poor response.
- For vendors, there are additional costs:
    - Customers who have experienced bad performance may not come back.
    - Lost sales, because the price per workload unit is not competitive.
    - Lost sales due to poor response time at an E-commerce site.
    - Free hardware upgrades to meet service level agreements.
    - Penalties for not meeting service level agreements.

Some Experiences

For about five years, I served as an in-house performance engineering consultant for a family of related software systems. My office was at their location, and each of the projects had the sense that I was a full-fledged member of their team, although each project funded only a percentage of my time. Subsequently, I spent another five years in similar relationships to a global collection of related and unrelated systems in locations ranging from California to Massachusetts to England to Poland. The ongoing relationships enabled quick assessment of architecture and design proposals, and quick identification of the root cause of performance problems that arose when deployment workloads changed dramatically. The cost to each project was about two to four weeks of performance engineering time per year on average, typically clustered at the architecture definition and performance quality assurance phases of a development cycle.

Customer Relations and Meeting Contractual Obligations

A communications system was sold with a contractual guarantee that it would meet security needs for 10,000 users when needed, but the feature was not developed for the initial system. When the customer needed the feature several years later, it was necessary for the vendor to implement it within the available resources on the now-obsolescent deployed hardware, or spend upwards of a million dollars to replace the deployed hardware at no cost to the customer.

Initially, a distributed architecture was proposed, because there was a belief that the central server in place could not handle the workload. A performance engineer worked with the architecture team one week after the proposal was first made, and showed that there was not enough bandwidth for the distributed architecture to meet requirements. Further analysis showed that a centralized architecture could meet the workload and response time requirements.

The extra cost of performance engineering for this project, was 6 months of a performance engineer's time, 1 day of rework for a module written by a new employee, and 4 months of a performance tester's time. In addition to supervising performance resource use measurements of the original and changed systems, and creating analytical and simulation models of the proposed systems (both architectures), the performance engineer was present at all design reviews, and worked closely with the tester to tune the operating system to handle the impact of a large increase in the software working set on a system whose physical memory could not be increased. The additional work of performance engineering had no negative effect on the original schedule. Had the original distributed architecture been implemented, 60 staff-months of rework would have been required; one week of a performance engineer's time was sufficient to prevent this.

The business need for performance on this project was extreme. The customer was very demanding in the area of performance. Had the vendor been forced to replace the deployed hardware, the receptiveness of the customer to sales of additional systems would have been greatly impacted. Instead, they were delighted, and a few years later, contracted for multi-million dollar integrated management system spanning all business operations.

Evaluation of an Outsourcing Proposal

A customer proposed to outsource the operation of an application complex running on 10 Unix servers to a third-party service provider who planned to port and deploy the software on a single mainframe running Unix. Two weeks of performance analysis showed that the proposed configuration was limited by the communications capacity of the mainframe's front-end processor, and that, with the proposed configuration, the mainframe could handle the work of only 1 Unix server, not 10. SPE saved the cost of a system port effort of $750K (6 months each for 10 people) that would fail when performance testing and/or deployment took place. The cost of the analysis was two weeks of application vendor performance staff effort. It saved a customer the cost of failure of an expensive project, and it improved vendor-customer relations.

Global Deployment

A proposal was made to deploy a mature software configuration management system for a global project. The potential customer was unconvinced of the viability of the proposal. An architecture review found that it would not be successful. The system architect and project manager were hostile to the finding, and continued to maintain that the system would work as advertised. A management-mandated performance modeling effort led to identification of a fundamental bottleneck with no workaround other than development of a system with a different architecture. The effort for the architecture review was about one week for each of 10 staff, half reviewers, and half project members. The performance modeling required a month’s work from a performance engineer, and about a total of a staff-month of effort spread over various project staff members. Early identification of a proposal that could not be successful enabled the upfront adoption of a viable alternative.

Cost Summary

$150K of SPE saved $1M in contract costs.

$4K of SPE saved $750K software porting cost for a system that could not handle the required capacity with acceptable response times.

$72K of SPE verified that an existing system was not viable for a new project in time to acquire or build an alternative system.

Conclusion

Relatively low-cost SPE efforts can provide significant risk mitigation. My experience is that the best cost-benefit ratio for a single SPE process task is to have a performance engineer work with the architecture or design team at the time that architecture proposals or changes are first considered. This should occur earlier in a development cycle than a review of a completed architecture. The additional cost incurred is the time of the performance engineer. The potential savings is the schedule and staff cost for a development cycle for an alternative architecture if the initially proposed architecture is not viable.