



---

## Buy or Build:

A guide for fuel retailers  
considering new fuels pricing  
system

## An efficient, automated and flexible fuel pricing system is a fundamental business necessity.

---

The question is whether you should build your own pricing system or buy one from a specialist fuels retail author. So, *what should you consider when making this decision?* Here are some questions to explore:

1. Is speed to market or the timing of your pricing system implementation important?
2. Do you want to take advantage of proven new technologies now and also in the future?
3. Do you want to benefit from system enhancements, capabilities and best practices from a range of other fuel retail organizations on a global basis?
4. Would your internal resources/system builder produce a longer ROI period than a specialist's off the shelf pricing solution?
5. Will they give you priority when functionality needs to be enhanced at a later date following the initial build?
6. Is the development, maintenance and support of pricing software your core competency? Can you gain access to support 24 hours a day, 7 days a week with clearly defined and monitored service levels?
7. Are internal development projects often underestimated in terms of effort, scope, maintenance and true cost?
8. Do you want the system's benefits to be clear and proven, thereby supporting ROI reliability?
9. Do you want the costs to be known and identifiable? The "true" cost of ownership of internal development is often difficult to calculate, manage, maintain and may even be disguised later. It is not just the initial cost of the build but the cost to maintain, support/fix, enhance as well as costs associated with training staff.
10. Do you want to eliminate the cost of "lost opportunities" due to delay or internal resource availability?
11. Would your internal resources/system builder struggle to maintain and update the system, especially when there are other IT projects competing for time or your supplier has larger projects to deliver? Critical to this aspect is retaining the talent that built the initial system - what if the key people involved in developing are no longer available?
12. Is it difficult to assess the real costs of custom data interface development and

maintenance? Are you concerned that the interfaces aren't flexible enough to accommodate changes in the data sources resulting from system upgrades etc?

13. Are you concerned about the degree to which the system is documented, so that any personnel change (within your own organization or the system builders' organization) does not leave you exposed or unsupported?
14. Is the ability of the system to adapt and react to changes in your marketplace a priority for you?
15. Do you want a system that is built along clearly defined quality guidelines?
16. Do you want a system that has been rigorously tried and tested, at length and in your market, prior to you taking delivery?
17. Do you want to take advantage of the system and its benefits as early as possible after your decision to proceed?
18. Do you want to benefit from high quality technical support AND many years' retail fuels pricing and industry expertise?

If your answers are “yes”, then you should consider buying a system.

## A few things to consider when building your own system...

Many processes around software development are poorly organized, resulting in rework, under-specification, unbalanced workloads, and other forms of waste that harm both productivity and morale.

Application development and maintenance waste includes:

### Overproduction and over processing

- Lack of understanding of project priorities, resulting in non-essential requests being fulfilled ahead of essential elements
- Unnecessary functionality being added simply because the capability exists

### Specification

- Specification is “loose” or lacking in detail
- Critical aspects of functionality are missed/wrongly detailed

## Rework

- Frequent changes in business requirements during development
- Application bugs

## Wasted motion

- Informal requests being made that are not tied to business priorities
- Ineffective prioritization of maintenance requests

## A large financial institution decided to assess its application development department efficiency. It discovered three key opportunities for tightening up its processes.

- Project requirement documents were vague. Developers spent time going back and forth for clarification throughout projects.
- IT had no standard procedure for obtaining the business's maintenance requirements. Developers would spend time going back and forth to the business for further information. This resulted in rework and missed deadlines.
- IT projects were not clearly or effectively prioritized.

## What do analysts say?

“One of the benefits of using a general but rigorous ROI methodology is adaptability. To ascertain total economic impact, or TEI, “enterprises with specific needs should place different weights on different categories,” says Forrester Research Principal Analyst Ray Wang in the executive summary of a research paper. “For example, an increased cost in a packaged application may be offset by a decrease in risk.

Likewise, the value of flexibility for an enterprise in a fast-changing and dynamic market may be worth more than [it would] for an enterprise in a more staid and predictable industry.”

Another benefit to the TEI approach is its applicability to a broad range of problems. “In general, enterprises should leverage TEI not only for evaluating vendor selection, consolidation, build-versus-buy, build-versus-rent, and upgrade/migration opportunities, but also for prioritizing entry points to service-oriented architecture (SOA) for packaged applications,” Wang advises.