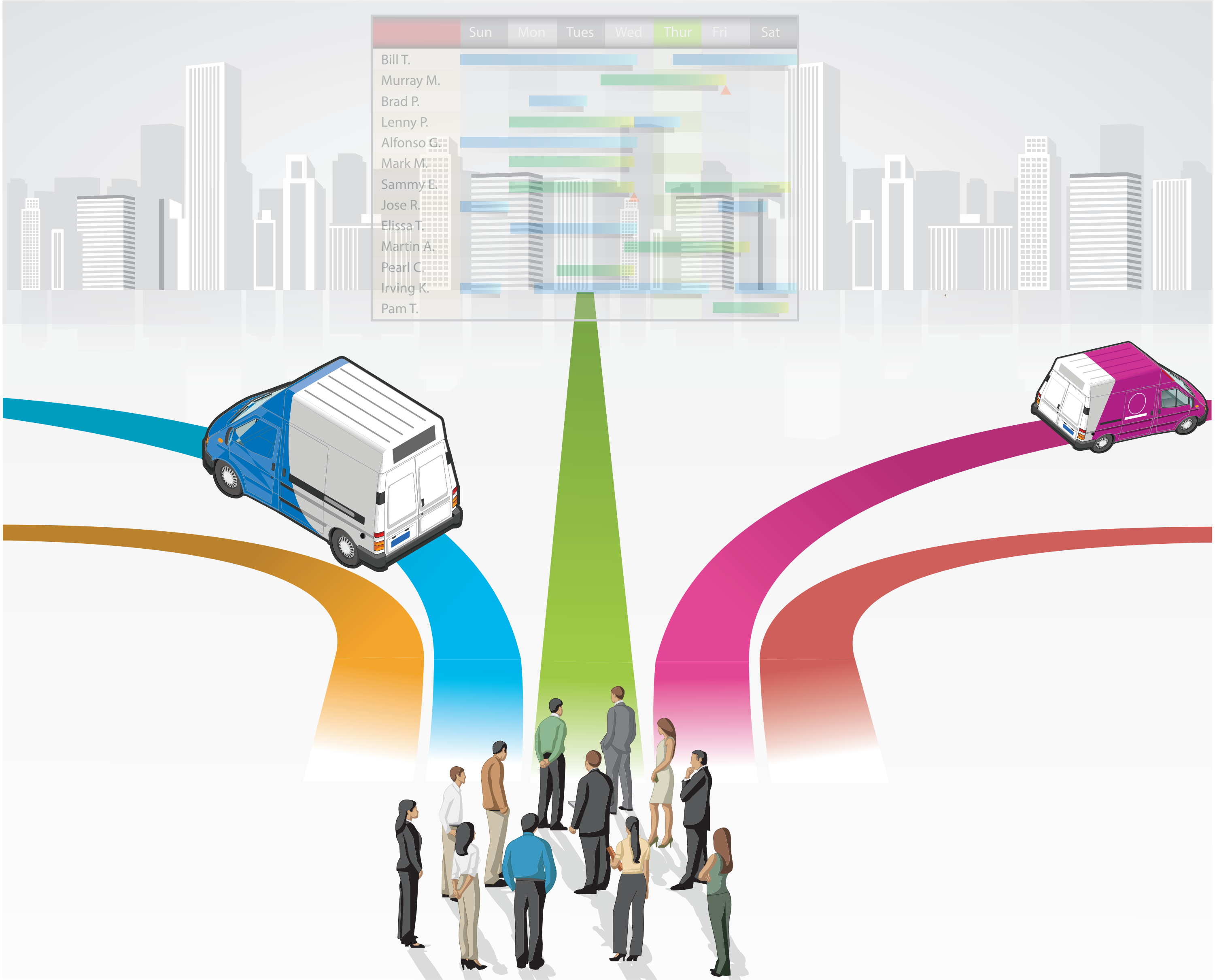
White Paper:

“Decisions, Decisions...”

Key Issues to Consider in Selecting a Field Service Management Solution



FIELDONE
ENTERPRISE FIELD SERVICE

Fall 2012

ver. 1.0

FIELDONE
ENTERPRISE FIELD SERVICE

Executive Summary:

Every company offering Field Service Management and Automation Solutions will impress you with a long checklist of features: schedule boards, dispatching & routing, work order management, automatic billing, and more.

How do you compare products and find the differences that should determine your choice?

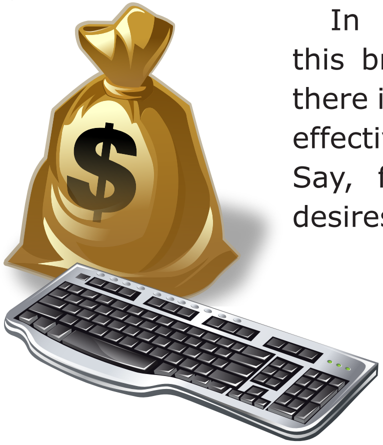
When making a life-changing purchase in your personal life, you always look beyond the basics. Considering a new house, you need to know a lot more than just how many rooms, total square footage, and the number of bathrooms. A new car? Engine size, mileage, color are *just* enough to get you started; next you dig into dashboard features, materials used in the interior, trunk shape and size, and the little things that often make *all* the difference.

When choosing a field service management solution to streamline your business's management of clients, jobs, tech, parts and finances— you need to assess each feature and capability to identify for strengths and weaknesses. Most importantly, the implementation of the product's new, improved processes have to *instantly* improve your business, showing rapid ROI— and not slow you down with a long installation process and steep learning curve, or require a radical change to the way you want to work.

The following are some of the key issues to explore as you compare products and their providers.

Development Costs

Today, the most common product implementation model in the market is the initial purchase of a base product with a preset, proprietary workflow and user experience scenarios, which is then customized and adjusted (within the parameters of the product's ability to adapt) to the customer's specific needs. Built into the vendor's financial models are significant charges for customization by a Professional Services team; the final bill is often two to three times the original cost of the base product. This is not trickery or bait-and-switch; it's explained clearly as "the way it is."



Aside from the obvious frustration these ongoing changes present to the customer, it triggers another predicament: Every change or customization in the future carries a new price tag. The manager responsible for authorizing the expense must be confident that even a general usability improvement clearly justifies the expense. The result is that usability-motivated adjustments that might decrease frustration, improve general productivity and free up time, but may not pass the "immediate ROI" test and thus be left behind.

So what *should* you look for to avoid both the costs and the dilemmas about making improvements? *Products built on industry-standard platforms with open architectures*. This creates wide availability of technical experts — both in-house and third party — who can develop (and troubleshoot!) on that architecture and improve your specific implementation, *without* necessarily locking you into the vendor's professional services. Microsoft's Dynamics platform is a good example of such architecture. There are hundreds of companies, employing thousands of developers who are experts in customizing applications written on this platform. A platform so widely deployed also makes it easier to find and hire *internal* experts who can continue to deploy and improve, keeping all work in-house. Needless to say, these platforms offer tools that allow customization to elements such as forms, workflows, dialogs boxes, reports, altering the look and feel of the interface itself in accordance with your needs.

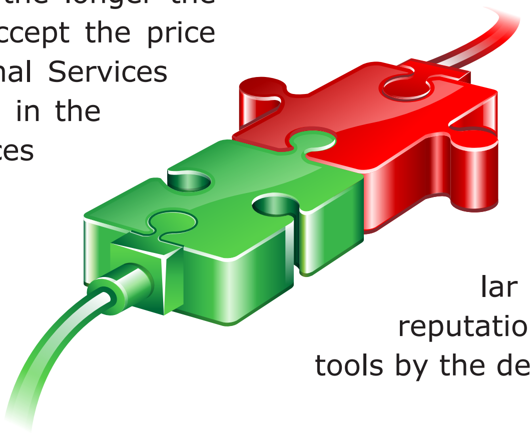
In addition to the availability of this broad community of developers, there is an even quicker and *more* cost effective option for adding functionality. Say, for instance, that a company desires to extend the platform, adding a full Project Management module, or perhaps a method to automatically manage Sales tax in the 6,000+ zones in the US. With a proprietary system, the vendor would be called in to build these features, generally charging a substantial fee to create the integration. As discussed, an open platform certainly means that the company can find an experienced and competitively priced developer to do the work.

But even more efficient is the availability of an *off-the shelf app that provides the needed functionality*, and which can be added in a plug n' play manner. With an industry-standard platform like MS Dynamics, for example, their Pinpoint Market offers literally thousands of such plug-ins that leverage the same data structures, reporting engines and other features already part of the platform.

The bottom line: Usability leads to ROI at every step of the process, and the easier and more affordable it is to improve on the user experience, the more often you'll see these modest improvements add up to measurable time saved every day.

It's a Matter of Time

The concerns about platform discussed above lead *directly* to implementation time. The more extensive the customization required, and the more limited the options for your internal team to make these changes itself, the longer the process will be. Even if you accept the price tag for the vendor's Professional Services team , the lead time required in the scheduling of these resources can often create delays. And though deployment *is* often planned in stages, the time to completion is often close to a year...and sometimes longer. Without the ability to adjust and improve on your own configuration, data import, workflow analysis and documentation, your reliance on the vendor may lead to a long, drawn-out implementation.



This returns us to the previous solution: an application built on a widely deployed and standardized platform. This means that if, for instance, about 80% of the application uses the platform's standard building blocks and offers easily customized components (changing settings rather than actual coding), there is only 20% remaining for third-party or in-house developers to tweak in a rapid-development environment.

Third-Party Integration

Let's assume you've got the configuration of the base product under control. Probably the most expensive, high-risk stage: connecting the new system to your existing data and applications. Porting all your data *completely* is rarely the first step, as you have invested substantially in the development of the applications, and your staff is trained in their use. Connecting these systems to the new product, which requires ongoing two-way data transfer, is by no means simple: At times, it's limited by a vendor's proprietary architecture and its often expensive "approved" connectors.

With an open platform, there are many ways to integrate into your existing applications and data, with some connectors built-in and some sold by third parties. And of course, the larger, well-known architectures offer a library of internal connectors: .NET, SOAP, XML, file import/export and a wide availability of popular middleware that have earned reputations as robust and dependable tools by the developer community.

Mobile Applications

Vendors of many enterprise-level mobile apps have made the tactical mistake of using HTML5 as their mobile platform. Their strategy is to create a single version of the interface, viewable on the variety of devices in their company through a browser window. While HTML5 is an excellent choice for the desktop, it is proving to be problematic in a mobile environment in areas of performance, security and limitations in the features it can access on the device.

Facebook founder Mark Zuckerberg confirmed this in a September 2012 TechCrunch Disrupt conference: "Betting on HTML 5 for the app is one of the biggest mistakes if not the biggest strategic mistakes we've ever made," he announced. "On IOS and Android, you can do so much better by doing native work, and we need to just do that." It's certainly easier and less expensive to use HTML5 rather than building a native app for each type of Mobile device and operating system. But the few companies that have decided to invest in native apps are providing a richer, more dependable tool for techs on the road.

For a more detailed explanation of the shortcomings of HTML5, see http://www.fieldone.com/products/overview/html5/

Central Mobile Management Console

Now that we've established the advantage of Native Apps over HTML5, the obvious question arises: Does this mean that when I decide to change a form, workflow, or add/change a dialogue screen, I need to create this changes in each of the various native apps I'm maintaining, then republish them all for distribution of the update?

In fact, some vendors require that you do just that, justifying it by saying that it's "the price you pay" for maintaining a high-quality native app that's truly optimized for each platform.

The ideal solution, however — one which is not too common, and truly exciting to see — is a central console where you make a single set of changes and the system builds it into each platform, including Android, IOS and Windows, according to its specific user interface paradigms. The changes are then published as new versions to your user base. This customer-centric approach keeps costs down, with time-to-deployment kept to an absolute minimum. Each version maintains the dimensions, UI and device features as appropriate, but can be enhanced and adjusted in a single action.

The Deployment Decision

There is no one-size-fits all approach to software deployment. Whether you prefer software hosted behind your firewall, the Cloud, SaaS, or Private Cloud, every company has priorities and realities that make some approaches more relevant, and some less. Even further, as dynamics within your company change, the preferences may change as well. For instance, a downsizing in the IT department might warrant a shift from in-house software running on a server bank that needs to be maintained, to a SaaS solution that relieves you of the need to update software and operating systems, service physical servers, back up data, and maintain security.

When talking with a vendor, ask about how many of these options they can offer, whether they offer substantial benefits (or downside) that are unique to each implementation, and how easy it is to migrate from one model to the other, should the need arise.

Security Architecture

Software companies selling platforms and architectures have developed broad, flexible, and scalable security frameworks to be used by virtually any type of application built on them. But developers of field service software sometimes create a more "business case specific" methodology to specifically address

needs of the sector. This works well much of the time, but often a company has roles that are unique or "hybrid", requiring specific types of access, permissions and restrictions. All systems will allow you to define a group, but individual users within a group may need special handling. Make sure that all your scenarios can be managed as-is, or the system is flexible to be easily tweaked to accommodate these requirements.

Automated Routing

The heart of your field management solution, Automated Routing, uses a complex combination of algorithms, logic, and user input to create the "ideal" matrix of instruction that guides techs from job to job with minimal wasted time. Obviously, some algorithms are more dynamic and effective than others, and some excel in speed while others focus more on accuracy. The important questions are:

- The application runs non-stop, drawing substantial computing resources; if I run this on a local server, will the hardware requirements be prohibitively expensive to purchase and maintain?
- Does the engine actually provide the very best solution, or in the interest of time, is it choosing the tech-job pairing from among the best options? The latter is not necessarily negative, as it allows for quicker assignments and reprocessing when new data arrives. It's simply important to understand what to expect.
- How easy is it to step in with human intervention to make adjustments to the schedule board? While full automation is ideal, there are often considerations the computer doesn't "know about"; these tweaks need to be included, manually, from time to time. A drag-and-drop system is the most intuitive and has become the standard for "managing the board". For more information about this challenge, see http://www.fieldone.com/products/overview/automatic-routing
- Is the system scalable, to handle usage substantially higher than today's? You have to assume that the system will need to make literally millions of calculations and scenario comparisons throughout the day. But adding 10% more techs does not mean the engine will work 10% harder; adding just a few tech or jobs will increase the number of calculations exponentially, so you need to make sure that the system will not slow down significantly (reacting in minutes rather than seconds) as your company grows.

In summary, comparison-shopping really has two stages: First, bring your checklist to the sales meeting or during your web research, and compare the lists of features that most vendors offer. Rule out the products that clearly don't match your needs. But once you have chosen a few finalists based on these lists, use this review to assess each feature more deeply, comparing and "scoring" the technologies chosen, the business models and the commitment to your ease of use.

FIELDONE
ENTERPRISE FIELD SERVICE

About FieldOne

Our goal at FieldOne is simple: To help your service company increase efficiency and productivity by leveraging technology that was built for your business's specific needs. We take great pride in knowing that we assist enterprises around the world, across a broad range of industries. Our clients share a similar desire to better manage the complex nature of their service organizations, and nearly everyone across the company can benefit from our software - from the owners to the service managers, the warehouse crew to the administrative personnel, and the schedulers and dispatch managers to the technicians in the field.

Founded in 2001, FieldOne is an innovative, secure and scalable service management system providing field service companies with powerful tools to streamline their business processes. From up-to-the-minute technician scheduling and status information to on-site wireless data entry, complete inventory control and automatic invoice creation, FieldOne is an easy to use, all-in-one software solution.

Our innovative, secure and scalable software enables our customers to spend less time entering and managing data, freeing up valuable time for revenue-generating work. With FieldOne, organizations can more efficiently handle all of their day-to-day activities without wasting hours on duplicate work. Most importantly, they can spend more time helping their own customers and growing their business.

We have had the privilege to help in industries including property management, HVAC, medical and diagnostics, IT and technology, mechanical, janitorial, landscaping, contracting, plumbing, electrical, roofing, irrigation and security companies.

For more information or to learn more about the benefits of FieldOne, or how our solutions can help you grow your business, please schedule a free demo or contact us.

1 International Blvd., 6th Floor   Mahwah, NJ 07495   1.866.918.8324
www.FieldOne.com   info@fieldone.com