Managing the Change Management of a new Timesheet System Deployment

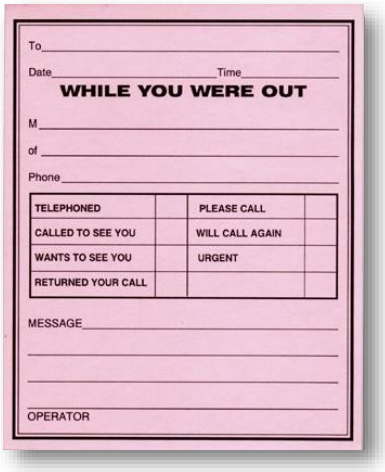
For more information contact:
HMS Software
189 Hymus, Suite 402
Pointe-Claire, Quebec H9R 1E9
Tel: 514-695-8122
Fax: 514-695-8121
Email: info@hms.ca
Web: TimeControl.com

Table of Contents

My name is Chris Vandersluis. I am the president of HMS Software, publishers of the TimeControl timesheet system. My career has been about managing the impact of change while deploying enterprise project systems. Over 35 years ago, our firm developed and marketed an enterprise timesheet system called TimeControl which was designed to integrate with both project management and Finance processes. TimeControl continues to be the cornerstone of our firm. As a result, I've had the opportunity to work with organizations in both the private and public sectors as they work through the challenges of deploying a change in their project management and finance systems. It's given me a unique perspective on how to manage such change in ways that are effective and to see the pitfalls of managing such change in ways that cause challenges. Over the coming pages I'll share some of what I've discovered.

Change causes upset

For anyone who has ever tried to implement any kind of system, they know this to be true. It doesn't matter much what the change is. Changing what people expect, will cause upset. Many years ago, I did a seminar for the project managers and business analysts of Bell Canada, one of Canada's most venerable telephone companies.

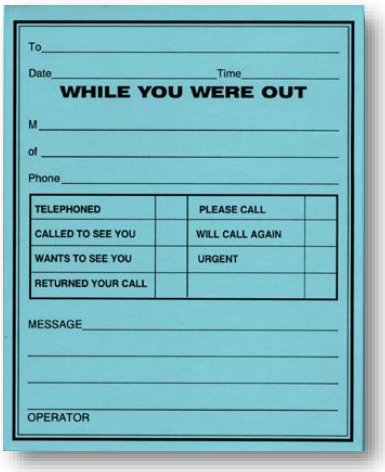


"Look at this message form," I said.

It's a common form, often found on the receptionist's desk. Pink is the most common color.

"Now imagine that I replace the pad with the exact same pad but in blue. I'd like you to imagine the type of responses you'd receive. Would they be happy responses?"

"Oh thanks. I was tired of the pink. This is so much better."?



Or would it be more likely that they'd be responses reflecting confusion and upset?

"What happened to our pink pads?"

"Why did the pads change?"
"We're all used to the pink pads."
"Who authorized the change from pink pads to blue pads?"
"People will be confused and won't return calls. We need the pink pads."
"When will the pink pads return?"

There was laughter throughout the audience as people imagined co-workers or even themselves responding in just this way.

It is easy for any of us to be upset with the change of the message form even though it is a small cosmetic change. The form itself is the same. The pages are just blue instead of pink. Now imagine that you are deploying a new piece of software for timesheet collection. This will affect every employee in the organization. This is a much more significant change than the color of a message pad. You should expect and plan for some people to be upset.

User Resistance. That's not a thing, right?

Around 2002, Microsoft evolved its Microsoft Project product into an enterprise project management tool with the release of Microsoft Project Server. I was very much in contact with the Microsoft Project Team at the time thanks to the integration of our TimeControl timesheet and Microsoft Project and accepted their invitation to join a Partner Advisory Council. It was a two-year commitment but they had me on the council for about 5 years so I was working with Microsoft directly on the releases of Project Server 2002, 2003, 2007, and 2010.

During the original Project Server release, Microsoft commissioned a study to identify the barriers clients experienced when deploying an enterprise project management system. The experiences were, of course, from other products but Microsoft was intent on discovering what features they could add or enhance in Project Server to ensure that their clients. I had the privilege of being at Microsoft when the results were presented to the Microsoft Project Team. Here were the top 8 responses:

Rate the following Enterprise Project Management System Implementation Challenge as "extremely" or "somewhat" challenging	
Culture Change (User Resistance)	100%
Lack of Knowledge on EPM System and Req's	100%
Executive Sponsorship	80%
Importance of PM to the Organization	60%
Long Term Funding	60%
Current State of PM Processes and Practices	40%
Technology Capabilities	40%
Competing Initiatives	40%

The reaction to this survey was silence. "We don't have user resistance here," was one person's final reply. That's entirely possible of course. But the resistance to enterprise systems like a project management system or a timesheet system is rarely about functionality. It's about a change in process. Just like Project Server, when we are trying to deploy a timesheet system like TimeControl across an entire organization, we are doing so in order to change a process. And that can generate culture change resistance in a hurry.

So, when planning a deployment of such a system, it is worth taking into account that the change won't be welcomed by everyone and, in fact, it might be resisted by some no matter how much better you think the new functionality is.

What are you trying to accomplish?

What business challenge are you trying to overcome. People are willing to change if it makes a difference. They're reluctant of change just for the sake of change.

In some of the interactions our firm has had with organizations interested in deploying enterprise project or enterprise timesheet functionality, we've had an odd exchange.

"We'd like you to help deploy this tool," the prospective client will say.

"Great," we reply. "What do you want it to do?"

"Um, don't you know?" the prospective client says, confused. "Aren't you the experts?"

"Well, we are the experts in this software," we answer. "But we have no idea what business challenge you're attempting to overcome with it."

Now, to be fair to the prospective client, sometimes this is because we are being contacted by someone who has been asked to find a consultant who can "deploy this timesheet system" or "deploy this project system" and while they personally may not have a clue what the business challenge is, someone in their organization does.

For those deploying such systems, it's important to think of the software you've chosen as a tool to solve something. If you don't know what problem you're trying to solve, then having the right tool may be difficult.

Sometimes when we ask what the problem is, the answer returned is, "My problem is, this software isn't installed. Come install it." That's a problem for someone to be sure. But it's not a business problem. In some organizations, a decision is made to deploy a software solution, but insufficient effort has been spent determining what it is supposed to fix. As a result, it's virtually impossible to determine whether the project is a success, a failure, or even complete.

Even in our own firm, we will speak to prospective clients who then talk to our technical deployment staff about what their goals are, and we find the list of requests are mired in functionality. "We want a button that does this…" or "We want the data to automatically fill in with the results." Our staff have been trained to extend this conversation and ask "If you were to get this function, what would the organization then do with the results? What business process does this serve?" Sometimes the answers send clients back to their business owner to find out and the result is a changed list of requirements.

The clearer the client is about what they want to accomplish from a business perspective, the more likely the success of the project.

People often have an agenda. (It's often not shared.)

In our line of work, we are often cautious about the agendas of people requesting things of us. It's fair to say that everyone involved in an enterprise deployment like a timesheet system has their own agenda. In some cases, people will have agendas that align with each other. In other cases, agendas may clash. With our own TimeControl, it is possible we encounter this more than some. The design of TimeControl is to fulfill multiple purposes and feed multiple processes at once. As a result, it has been made to be incredibly flexible. So, when someone says, "Can we do this?" The answer is almost always "Yes. But should you?"

For someone deploying an enterprise system, it is important to determine what people need to accomplish to be satisfied. We work on that in design sessions. Clients often describe what they want in terms of the functions they want to see. This is so common that grids of functionality are usually the way our sales people first encounter the needs of the client.

A description of a function, though, is a description of a solution not the problem. So we take that further when we do a preliminary meeting to design the configuration of the client's system.

"What will this solve if we configure your system to deliver that function in the way you are describing?" we will ask.

When there is a requirement listed, we will be quite intent on having the author of that requirement present to explain what business challenge they are trying to overcome.

It is common in these meetings to find that different people have different business challenges and some of those challenges might conflict. Conflicting business challenges aren't a problem. Once they're transparent, they're just something to deal with in the configuration.

"We need to do approvals only once a week," one person from HR might say, "because that's all the budget of time for approvals we're willing to allocated.

"We need to see the timesheet data as it's entered every day," someone in project management might say. "So, a weekly cycle doesn't work for us."

"How about a weekly approvals cycle to make sure HR gets what they need and give the project managers visibility into timesheets in progress?" might be our compromise suggestion.

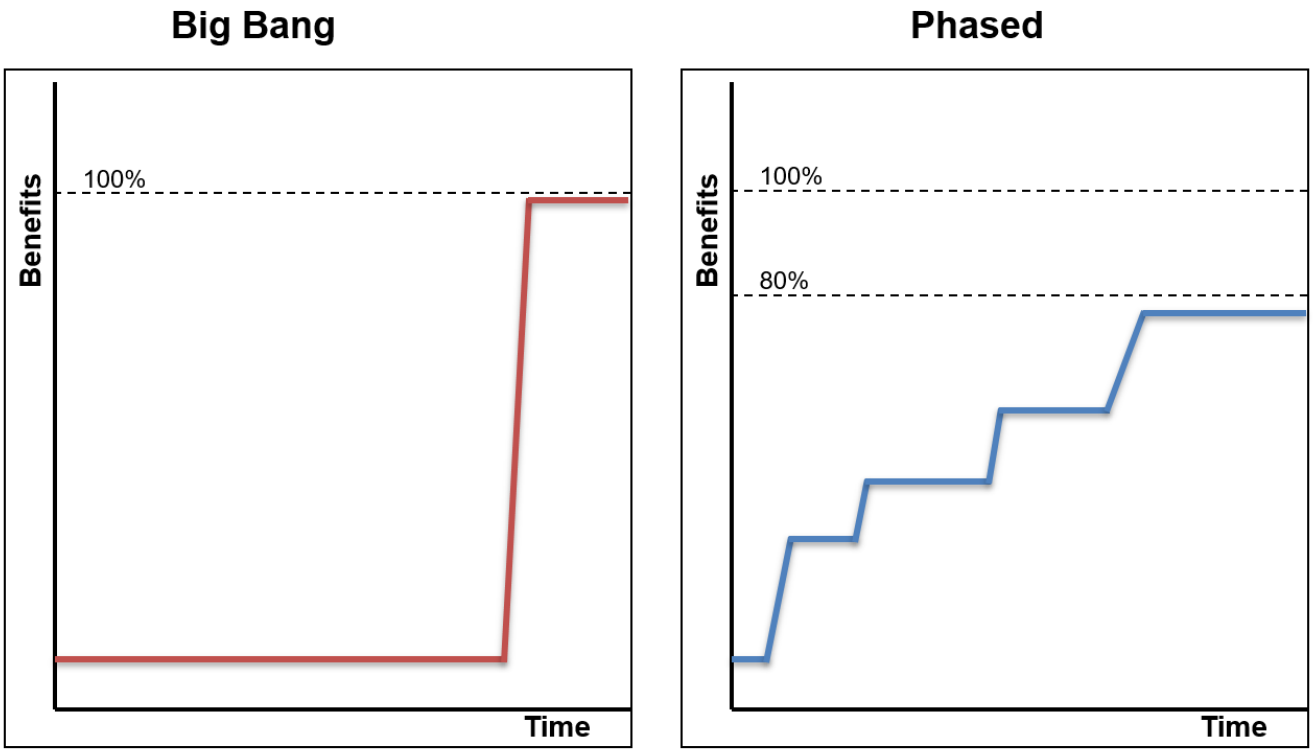
It can also happen that more subtle agendas are revealed in such meetings. "Why don't we just keep our old system?" someone who is responsible for the old system might ask. "Why don't we write our own system?" a programmer from IT might ask. It's good to get these agendas into the open when possible and at a minimum, keep the bias in mind as you do meetings and the project evolves. Trying to ignore hidden agendas or shout them down is rarely effective.

It's key to keep the interests of the stakeholders and participants in mind as the project evolves. Making sure the essential elements of what people need get taken care of can make for a highly successful project.

One aspect of an enterprise system implementation is the roll-out schedule. There are two fundamental approaches to consider. One approach is defined by having all the requirements established and validated before starting then doing the work to code, configure, test and train and then, on a given day, release the final product for use. We'll call that the Big Bang plan. A second approach is defined by finding smaller chunks of work that can be defined and releasing those in a phased manner.

Some may say, "Oh, that's Waterfall vs. Agile" and that may be true in some situations but the approach I am describing could be delivered using any planning technique and we've been talking about this approach since long before Agile was a term that anyone had heard.

Let's consider the following diagram.



The Big Bang plan

In the left, the Big Bang plan has us keeping the system we'd like to deploy unavailable until such time as we deem it 100% complete. There are advantages and disadvantages to this plan.

First, on the advantage side, the chances that the system will be released with 100% of functionality is quite high as that is our basic plan of deployment.

On the downside, the benefits to the organization of the new system won't be available until 100% of the defined needs of the organization are met.

There's more bad news in the Big Bang philosophy. For every day the project is still in progress, something can happen to the basic plan. And by something, I mean something bad.

There could be a change in management and the new management might have no desire to support the project. There could be a change in requirements as other people in the organization think of other ways to solve their business challenges. There could be changes to the company itself as it is sold, merges, or changes structure. There could be changes in the marketplace as a competitive product or idea emerges. Or changes in the economy where the business need for the solution being deployed changes.

Plus, there's one more thing that is rarely mentioned by advocates of Big Bang thinking. Once the solution is delivered and users start using it, there is a 100% chance that someone will ask for changes. The ability to ensure that all requirements are met in the absence of seeing and using the solution envisaged is very challenging.

If the project can't get completed, no one gets any benefit from it.

Phased Approach

We've long advocated a phased approach in projects. This isn't a unique approach to project management or even unique to the software world. In construction, this kind of thinking is usually termed "Design/Build" because the design isn't all complete before you start to build.

In a Phased Approach we try not to think of the final product except insofar as we must provide for anything it might need in the architecture. Think of a Design/Build construction project. If we are building a 10-story building and we're just not sure what the building will consist of after the first 3 floors, we still need to provide a foundation that will support all 10 floors. The design will include provisions for the 10 floors even though we're still designing the top 7.

In a deployment of a project control system like TimeControl, we ask the client to think about a couple of things, first what are the overrearching business challenges that this timesheet will resolve. Second, what is the least we can start with.

Specifically, we ask, "Please tell us the least we could deliver, the delivering of which would cause a positive return on investment every week."

That's quite different than Big Bang. We need to know how little of their desires need to be met to be sure that the value every week is larger than the cost and annoyance every week. If we achieve that, the system will be glued into place for a long time. No one throws something out that is producing more value than it's costing.

Then we start thinking of a Phase 2, Phase 3, Phase 4, and so on.

There is always a desire from some when thinking about this to see how they can have everything but even the harshest critics of the approach admit that it has a lower risk than the Big Bang plan.

There are advantages and challenges to this approach also. On the negative side, the chances that we never reach 100% of the original envisaged requirements is high. At some point along the way, let's say for argument, the 80% mark, the client can say, "I think we've got enough now from this system, let's just stop it there."

On the benefits side though, a return on investment can happen very, very quickly. The client can start to reap the benefits of their investment even while they're still working on configuration and design of other components. Also, this type of approach is quite friendly to the requirements changing. In fact, the allowance for change is built-in. As each phase begins, there is a reconsideration of what to go into the next phase and items that had never been considered before can be introduced.

The plan still needs, of course, an overarching architecture and a guide to keep ensuring that the system continues to deliver on the business challenges the organization has.

Which approach is best for me?

There's never a right and wrong answer to this. Clients should adopt the approach that is most appropriate for their situation.

To be sure, there are some projects that don't lend themselves to a Phased Approach and must have a Big Bang notion. Think of systems designed for health care or for banking or defense. All of these must be 100% complete before they can reach the user but even in these situations, there are often opportunities to phase the work although perhaps in bigger phases.

Other organizations can take the Phased Approach too far and disconnect the people making the phases and making the work from the core business challenges that started it in the first place. Agile organizations must be on guard for this on a regular basis.

We often encounter two steps that clients lobby for before deployment. They are most commonly known as a “Proof of Concept Phase” and a “Pilot Phase.” Some people use these terms interchangeably, but they are quite distinct. We’ll tackle both of these below.

Proof of Concept

Clients will often ask a salesperson to help with a “Proof of Concept” prior to committing to an enterprise system like TimeControl. Before we can reply, we need to have the client define what they mean by the term.

“What concepts do you need proven?” we will ask.

That usually brings the conversation to a grinding halt. To be sure, sometimes clients have an actual technical concept that needs proving and that’s usually because they’ve had difficulties with that technical aspect in previous systems. Perhaps they need to know the system will be responsive from any of their geographic sites around the world. “Will the users in Asia get the same performance as the users in the US?” we might be asked. Sometimes there is a need to show that the system can function under certain volumes of data. “Can you prove that the system will work well with 1,000 users on it?” These types of technical proofs of concept are quite simple to manage.

However, what some people want is to know with certainty what the system will be like once it is fully purchased, fully configured, has their own data in it, once it is integrated into other systems and, once everyone is trained and once it is completely accepted by their users. They’d like to “Just see what it would be like once it’s fully implemented”.

Moreover, this desire to see what the successful project will look like will have to occur without funding, without a commitment by management to engage in the process and without allocation of key personnel to make this part of the project successful.

The person might have no problem understanding that once purchased, it will take several weeks of effort to complete the configuration, integration with other systems, data loading and user acceptance. Yes, they often have great difficulty understanding why all of that can’t be done in a couple or a few days of effort for their needs prior to commitment.

We often point out that if the company isn’t ready to engage its resources in the proof of concept phase with design time, management effort, budget and the allocation of key resources, then they’re unlikely to experience much success in seeing what the end product will look like.

For some clients this means going back to management and we will work with them and their management team on figuring out what will make them comfortable enough to advance. In many cases, just being able to talk to existing TimeControl clients with similar business situations is enough to give management the confidence to move forward.

Pilot Phase

A Pilot Phase is a phase of deployment prior to going live with all users to ensure that the requirements that will be needed on the go-live date are met. When this is what clients mean by a Pilot Phase, then we're in agreement.

We generally thing a Pilot Phase is a key component of a successful rollout of TimeControl. It is something that needs to be budgeted into the schedule between user acceptance and go-live. We expect all the integrations, configuration, training of key personnel and data loading to be complete. The Pilot Phase is often with a subset of the entire database and a small section of highly motivated personnel who will have been given time dedicated for the Pilot to do reporting, follow up of any perceived discrepancies, meetings and so on.

A Pilot phase can last a couple of weeks to a few months depending on the sensitivity and complexity of the deployment. Often, running one or two payroll cycles or one or two project update cycles is a popular duration for the Pilot. Or, if the system will be used for billing, running one or two billing cycles.

One of the advantages of a Pilot Phase is that actual users are on the new system and the organization is starting to receive benefits from those users on the system. This can help motivate other users who hear about this new system and want to get on board. Also, the users in the Pilot become a key internal resource for the company because they will have had extra weeks or even months of experience with the new system and can help guide new users into how to use it most effectively.

Also, a Pilot Phase is designed to identify anything that may have been forgotten during the configuration and design. If the design was done well, it's rare to come across major items during this phase but reasonably common to come across minor items that can be corrected quickly and put into place prior to the roll-out.

When your enterprise timesheet deployment project is getting to the pilot or roll-out phase, you may encounter a combination of different attitudes of who wants to be first. You will have to determine which groups are most likely to have either your Pilot Phase or your earliest live roll-out phase be successful. Here are just some of the groups you might encounter:

We're eager but we are inexperienced

You might find one group that is very intent on going first but when you investigate it turns out that they are intent because they have no solution at the moment and want one, but they also have no internal processes that exist for a timesheet roll-out. This can be a challenge.

We are eager, but we aren't really

There may be some groups who are intent on proving that the system suggested will never work. They've probably been quite vocal all along the process and now, suddenly seem super eager to be the first users onboard. There may be some temptation to say "We'll work on the toughest audience first" with a notion that if we can get this group on board, all the others should be easy. This is a temptation to be resisted. A group that has a list of why the current solution won't work for them will be the hardest to enroll and to convince to dedicate themselves fully to the project. Once other groups are successful and the full deployment of the system inevitable, these harder groups will be much easier to come on board and as a bonus, other users will help to support them.

We're willing but we're already too busy

Picking a group that really needs the efficiency of the new system will be tempting because of the benefits that will ultimately be delivered, but in the initial stages of a deployment, it's much more important to make sure that people have enough time to learn and take advantage of the new system as it becomes available.

We are brand new and we've been told we're the first on the system

This kind of group can be tempting because they have no prior knowledge or experience with prior corporate systems. However, this kind of group is also a big challenge because they have no prior knowledge or experience with prior corporate systems. It's tough enough to be onboarding to a new firm and learning how things work without being used to see how a new enterprise system like a timesheet will fit in. It's far better to have people in the initial group that are familiar with the corporate systems and culture.

We are a team assembled for this purpose

This might be the best option for an early deployment team. A group that is cross-functional and cross-department can, once their deployment is complete, become a key resource that is redistributed around the organization with their newfound knowledge. Once a Pilot Phase is over and the roll-out begins, these people can then act as ambassadors of the new system as well as key points of contact for new users to get knowledge from.

Overcoming challenges and dealing with sinkholes and pitfalls

It is perhaps common nature to think that if we organized our plan properly in advance; if we crossed all the t's and dotted all the i's, then the project should just come to a conclusion without difficulty. That's not always the case however. Any project can encounter an unexpected sinkhole or pitfall along the way.


Sinkhole in Tampa, FL, circa 2013

There are many challenges that enterprise deployments can encounter. They can be due to user resistance or hidden agendas or insufficient resources or a change in personnel or landscape for the project.

When we're asked to come into an organization and help with unexpected challenges we always start at the same place:

- ❑ Where is the project charter?
- ❑ Where is the project schedule?
- ❑ Where is the project budget?
- ❑ Who is on the project staff?
- ❑ Who is the business owner?
- ❑ Who are the management sponsors?
- ❑ Where is the design of the system?
- ❑ What are the objections and who are they from.

As you might guess, on a regular basis, we find the project personnel who are working on the project don't have many of the documents they would have insisted for any of their other projects. When we try to retrofit a project charter to the existing situation and get management buy in after the fact, we're usually in for a difficult time.

Here are some common pitfalls we've encountered in the past:

- ❑ No schedule or an unrealistic schedule
- ❑ No sign-off criteria that will mark the successful end of the project
- ❑ Requirements that are outside the scope of the project
- ❑ Insufficient attention to the change in processes that will result
- ❑ A belief that technology, once installed, will fix everything
- ❑ A lack of executive sponsorship or insufficiently long sponsorship
- ❑ Unwillingness or inability of personnel to change
- ❑ Insufficient budget for training, configuration and post-deliver changes

To help protect the project against unexpected sinkholes and pitfalls, here are some recommendations:

Treat your project as a project

Start by treating your timesheet deployment project as a project with all the standards for projects you use for everything else. Just like with any project, make sure you have

a schedule, an executive sponsor, a business owner, a business plan, a schedule and a budget. Just making a plan like this is a huge first step to making sure your project has a beginning and an end and gives you a much better chance of making it successful.

Executive Support

Sure, someone in management will sign off on the purchase order of the new system, but an enterprise system like TimeControl will result in a change in corporate culture. Who from the executive suite is signing off on that? Will they be with the project until its completion and understand completion isn't defined as "Software installed" or "Software subscription started"?

Have success criteria

It's surprising to us how often this is missing and how surprised our clients are that we ask for it. "What will constitute a successful project?" we ask. Hearing "the software is installed or "our subscription is active" always has us return to ask about the business challenges being met. If there are no business problems, then we are challenged to know if what the solution should be. When business challenges are identified then we are all working towards having that problem resolved and the project deemed a success. A project manager, even one will little experience, who insists on the business criteria to be defined will make them look like a genius.

A realistic schedule

Having a schedule that goes from the start of the design all the way to project success is a key success factor. Often we see just a deadline with no interim milestones and a lack of appreciation of who in the organization may be affected.

"This looks like an export to the payroll system," we might say. "Who from Payroll will be sitting in on the design?" A lack of an answer is a red flag.

"This will change the current cadence of timesheet collection," might be a question. "Who from the existing system is on the team?"

Make sure your schedule includes all the parts of the organization that is affected.

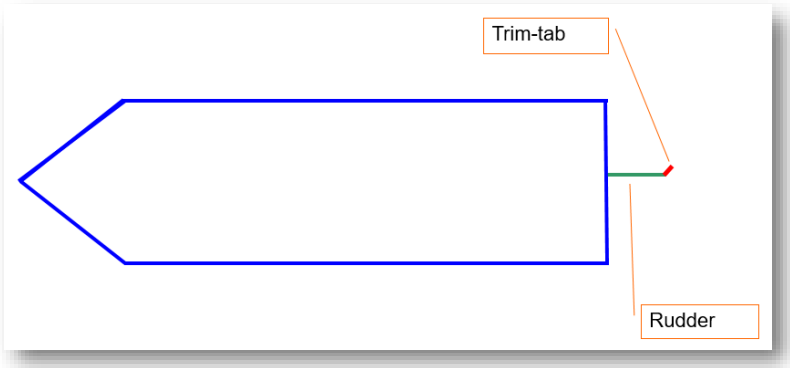
Communicate, Communicate, Communicate

An enterprise system deployment like TimeControl will ultimately affect almost everyone in the organization in some way. An absence of knowledge in advance can have people worrying the worst. You can help by communicating the progress of your project and how people will encounter it as the project progresses. There are so many avenues to do this that it's impossible to not be able to find at least one. Corporate newsletters, blogs, seminars, webinars, podcasts, slide shows and so on can all be useful criteria. This kind of communication along the way can also reveal potential concerns prior to roll-out that can be addressed and overcome by the team long in advance.

His epitaph reads, “Call me Trim Tab.” Buckminster Fuller was an architect, an inventor, and an author, but he is perhaps most appropriately known by project managers as a systems theorist. Bucky Fuller is perhaps best known around the world for his design of the geodesic dome, but when you read his theories on how to change a system, it makes anyone involved in project management take notice.

“Something hit me very hard once, thinking about what one little man could do,” Fuller reported in a 1972 Playboy article. “Think of the *Queen Elizabeth*—the whole ship goes by and then comes the rudder. And there’s a tiny thing at the edge of the rudder called a trim tab. It’s a miniature rudder. Just moving the little trim tab builds a low pressure that pulls the rudder around. It takes almost no effort at all. So I said that the little individual can be a trim tab. Society thinks it’s going right by you; that it’s left you altogether. But if you’re doing dynamic things mentally, the fact is that you can just put your foot out like that and the whole big ship of state is going to go. So I said, ‘Call me Trim Tab,’” - (Farrell, 1972).

So, let’s think like Fuller of a ship the size of the Queen Elizabeth. The rudder of the Queen Elizabeth weighed a staggering 140 tons. That’s just the rudder. The force needed to push that massive rudder to the left or the right would have required hydraulic engines that would have been massive.



Faced with this dilemma, marine engineers came up with a solution. “Forget the ship,” they say. Pretend we only have the rudder to move. If we move the entire ship with a rudder, can’t we use a smaller rudder to move the rudder? Thus, the trim tab is born.

I don’t think Fuller cared much about the marine engineering of rudders, despite his time in the Navy. But Fuller cared very much about leverage. About how the tiny energy of one person might be able to affect an entire planet. The trim tab of the Queen Elizabeth could have been pushed with the force of a single person and with that force, he or she could have that massive ship turn in any direction he or she wanted.

This trim-tab effect can often be seen in change management projects. We ask our deployment experts to be on the lookout for a trim tab. What tiny change in practice or procedure would cause the entire ship (in this case the behavior of the organization) to change?

A battle plan lasts, until contact with the enemy

It is one of my favorite expressions in the project management industry. And it seems to ring true in almost every project. It's critical to plan. But to think that the plan will last unmolested until the end of the project is naïve. Life happens to plans. So, be flexible. Listen to the words of Napoleon Bonaparte:

A battle plan lasts, until contact with the enemy.

- Napoleon Bonaparte

Easy to use Interface

- Full web-based browser interface with multiple browsers supported
- TimeControl can be implemented within a SharePoint interface or a Microsoft Project Web Access interface
- Scaleable user profiles facilitates use for data entry users yet provides full functionality for administrators
- Multilingual with multiple languages included
- Unlimited charge codes displayed in simple, hierarchical drop-down lists
- Unlimited free-form notes for each line item and each timesheet
- E-mail-enabled. E-mail messages sent for system notices such as rejected timesheets or missing timesheets
- Schedulable E-mail notification for missing or unapproved timesheets.
- Predefined timesheets based on resource assignments from the project management system or by user input

Robust Architecture

- Open database architecture; support for Oracle, Microsoft SQL Server, Sybase and MySQL databases
- N-tier architecture makes system scaleable for 10 to 100,000 users
- Unlimited rate codes per employee
- Field-level security. Make any field visible, value read-only, or invisible
- Complete redefinition of every field label
- Complete auditability of timesheet data
- User-defined fields on every table
- Add pop-up data validation for each user-defined field
- Allows charges to be linked to a specific project or project-independent
- Multiple overhead charge types
- Filter charge codes, projects and rates visible to any employee

Web Interface

- MyTimeControl™ home page dashboard gives extensive and customizable dashboard information to employees

Mobile App

- The free TimeControl Mobile App is available on Google Play for Android devices or the Apple Store for iOS devices

Approval Process

- HMS's unique *Matrix Approval Process for Labor Actuals*™
- Unlimited automatic Validation Rules are user definable, flexible and can be applied globally or to any group or even an individual
- Unlimited manual validation levels in which each employee can have a unique approval routing
- Project Managers or Account Managers can preview and redistribute hours prior to linking with a project management system or exporting to Finance

Links to Project Management

- Direct integration with popular project management systems such as Microsoft Project, Project Server and Project Online and Project for the Web, Oracle-Primavera Pro and Primavera EPPM, Hard Dollar, VersionOne, BrightWork, ARES PRISM, JIRA and Deltek EPPM
- Supports multiple project management systems and multiple versions simultaneously
- Customizable import/export function to interface with virtually any finance or ERP system including SAP, Oracle, PeopleSoft and Microsoft Dynamics
- Interface can be integrated directly into SharePoint, Microsoft Project Web Access or stand alone

Vacation Requests

- **TimeRequest**™ module allows vacation, personal or other leave time to be requested
- TimeRequest allows multiple levels of approval
- TimeRequest automatically populates future timesheets with approved time off

Flexible Reporting

- Excel-like reporting format allows output to any Windows-compliant printer or reports can be saved as Excel, XML or HTML files
- Reporting Wizards allow an unlimited number of reports to be created and saved for later use
- Unlimited levels of data selection, filtering and sorting
- Drill Down Analyzer provides instant ad-hoc analysis of data at any level

Expense Reports

- Users can enter non-labor costs on their timesheet
- Unlimited number of expense items per timesheet line item
- Expenses can be tracked back to a project management and/or finance system

Government Compliance

- Complies with requirements for DCAA, European Time Directives, FMLA, the California Wage Laws and Sarbanes-Oxley

Hardware Requirements

- Server:
  - Internet Information Services
  - MS SQL Server, Oracle or MySQL database
- Browsers:
  - Chrome, Firefox, Safari, Edge, Mozilla and Internet Explorer
- TimeControl Mobile App:
  - Android phones and tablets, Apple phones and tablet

Available Online or On-premise

- Available for purchase for an on premise implementation or Online as a subscription in the cloud

Engineering/Construction
Andritz
CANAM
Koch Business Solutions
Keppel Floatec
JRI Engineering (Chile)
Aegion
SCX (UK)
Foster Wheeler
ESI Inc of Tennessee
Karridale (Australia)

Manufacturing
Hamon Deltak
Tennant
Volvo-Novabus
Esterline Advanced Sensors (UK)
Kirchhoff Automotive
Progress Rail
Enerkem
Scheidt Bachmann
ASL Environmental Sciences

Defense / Aerospace
Aero Info (a Boeing Company)
CAE Electronics (Australia)
General Dynamics (Australia)
Safran
PAL Aerospace

Government
USCG (United States Coast Guard)
SFMTA (San Francisco Mass Transit)
AEMO (Australia)
Alachua County
RBQ (Régie du Bâtiment du Québec
City of Rocklin
Ross Valley Sanitary

Technology
AMD
EXFO
R2.ca
Ricoh
March Consulting
Accedian
Videojet
Scisys (UK, Germany)
Ascendant (Bermuda)

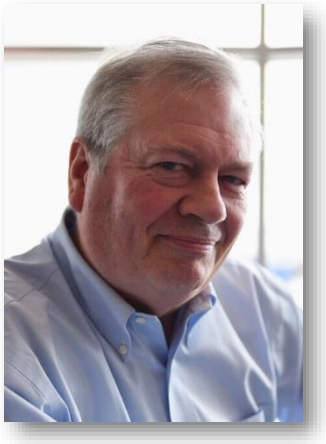
Financial
Centre de Recherche Informatique de Montréal
MCAP
Residential Mortgage
Value Recovery
Vancity

Health/Pharmaceutical
Conduent
Sandoz
Zoetis

Conservation
Kawartha Conservation
Minnesota Land Trust
Conservation Ontario

Other
Kelly Services
Reebok-CCM
Pontoon Solutions
US Properties

About the Author



Chris Vandersluis is the president of HMS Software. HMS Software, based in Montreal, Canada has been a leading provider of project management and enterprise timesheet systems and services since 1984. HMS Software's TimeControl® is recognized around the world as the most flexible project-oriented timesheet system.

Mr. Vandersluis has a degree in economics from McGill University and over 30 years' experience implementing enterprise timesheet and project management systems. Mr. Vandersluis spent five years on Microsoft's Enterprise Project Management Partner Advisory Council and has worked with Oracle-Primavera and Deltek on their project management systems.

Mr. Vandersluis' has been published in a number of publications including Fortune Magazine, PMNetwork magazine, Microsoft's TechNet and is the author of the popular project management blog EPMGuidance.com.

Mr. Vandersluis has taught Advanced Project Management at Montreal's McGill University and has been a member of PMI since 1986.

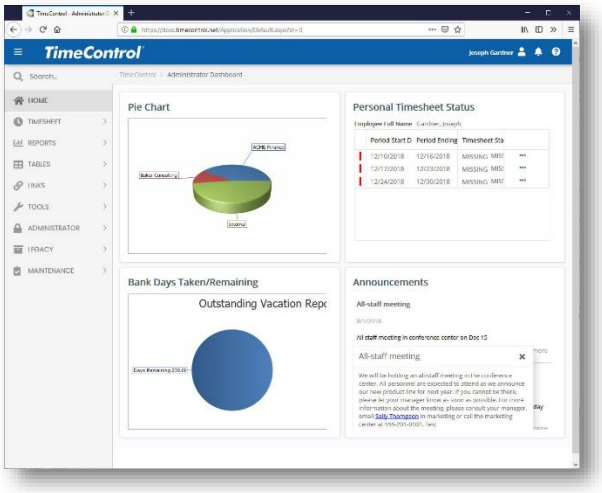
Mr. Vandersluis lives in Tampa, Florida. He can be reached at: chris.vandersluis@epmguidance.com or at chris.vandersluis@hms.ca.

HMS Software, is a leading provider of enterprise timesheet and project management systems.

Founded in 1984, HMS Software's expertise in implementing enterprise project-management and enterprise timesheet systems is recognized worldwide by some of the world's best-known organizations. HMS's signature product, TimeControl, an enterprise timekeeping system designed to serve the needs of both Finance and Project Management, is distributed worldwide through an extensive list of distributors and dealers located on every continent with representatives in the US, the UK, Australia, Mexico, Europe, Asia, South Africa and the Middle East.

HMS Software's client list includes some of the world's leading corporations in the telecommunications, IT, finance, engineering, defense/aerospace and government sectors including such organizations as Aegion, CANAM, CAE, the City of Montreal, Electromotive Diesel, EXFO, Foster Wheeler, Kelly Services, the Government of Quebec, Pontoon Solutions, Reebok-CCM, Rolls Royce, Sandoz, SEFA, Volvo Novabus, Zoetis, and hundreds of others.
HMS maintains offices in Montreal, Quebec and Toronto, Ontario.
For more information about HMS, please visit www.hms.ca.

About TimeControl



First published by HMS in 1994, TimeControl has been adopted by hundreds of clients and hundreds of thousands of users around the world. TimeControl is designed to serve the needs of both project and finance simultaneously. It allows an organization to use a single timesheet for project tracking, time and attendance, time and billing, HR tracking, R&D Tax Credits, DCAA and project costing instead of having to deploy many timesheets to serve these needs. TimeControl is available both as a subscription as service and for purchase for an on-premises implementation. TimeControl's architecture is flexible and extensive supporting numerous databases such as Oracle, Microsoft SQL Server and MySQL, multiple browsers such as Internet Explorer, Firefox, Safari and Chrome and even includes a mobile interface for Smartphones
For more information about TimeControl please visit: www.timecontrol.com.