HAMMERTECH

In partnership with

PAYAPPS
CONSTRUCTION PAYMENT APPLICATIONS
A ZUUSE SOLUTION

A practical guide to construction software implementation

Introduction

First of all, congratulations!

You've made the decision to purchase a new piece of software that is designed to help you improve the way you work, save time and money.

We generally adopt technology with the intention to improve processes and margins. But buying software isn't just the solution by itself – how you implement, communicate and train all of your stakeholders has a huge impact on the success of your software rollout.

To set your team up for success, not just in the short term but for the long term, successful usage and uptake is the key.

In this guide we'll provide practical tips on what makes for a successful software rollout: from assessing if a solution is right for your organisation, through to stakeholder communication and training. We'll help you avoid the common pitfalls.

So, let's jump in.

Stage 1 – Pre-Purchase

Before you even make a purchase what are some of the things you need to consider to prepare for implementing software?

Step 1: Identify what are the actual problems that the software should be addressing.

High level:

- System capabilities vs your requirements – get these aligned and clear.
- How big of a change will this be from your current way of working?
- Does everyone know your current processes or will they need to be trained along with the new software?
- What is your current technology setup i.e. connection on site, age of laptops, access to tablets etc?
- Implement on a couple of projects first, to get the process correct, before rolling out.
- Make sure the people that will be implementing the software, are part of the decision making process.
- Focus on needed requirements, not nice to haves.

Step 2: Understanding your own processes first:

Making preparations for implementing software is a great place to start, because if you get this right, then you set up for a smooth implementation.

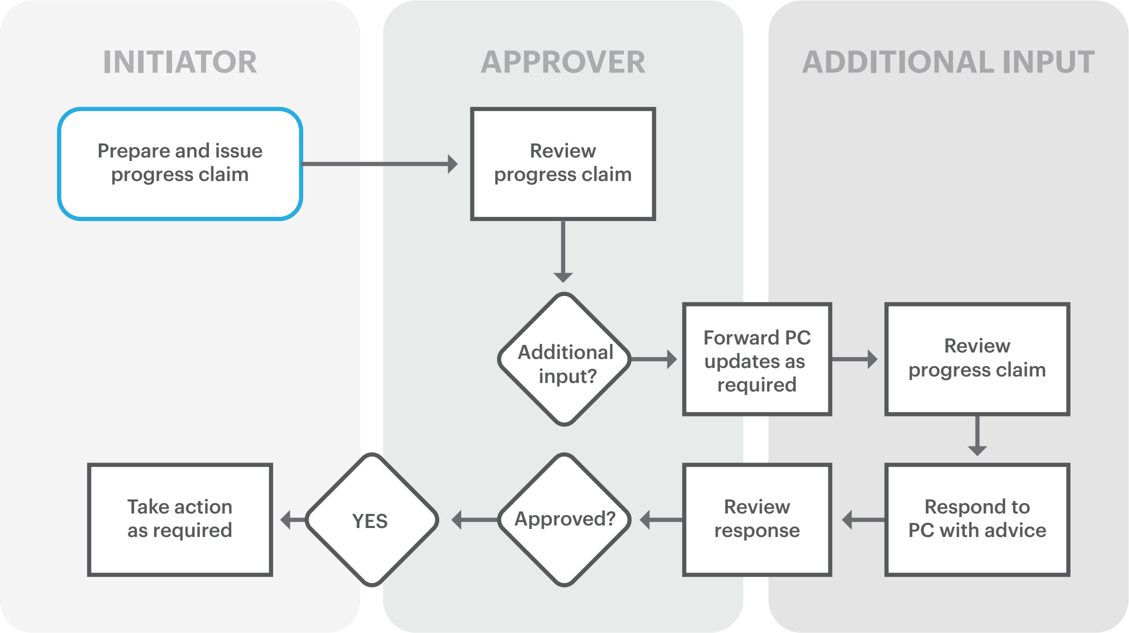
Before you go to a software vendor it's important to understand:

- How you are currently doing things?
- What are the pain points?
- What are you not currently able to achieve that you're hoping to achieve with software?
- What is currently working well?
- Does everybody know the current processes and are they using them?

During an implementation, it can be shock to discover onsite teams don't even use the current processes. So discovering this early on can be something that is rectified during the whole training process during implementation. Understanding this upfront is important.

Mapping processes doesn't have to be an onerous task. It can be done literally 'on the back of a napkin" or through a more thorough process mapping series of workshops. A process map visually shows the steps of a work activity and the people who are involved in carrying out each step.

When mapping a process, you simply draw a box for each step and connect them with arrows to show a flow. A basic process map would look like this;



Nintex offer a range of free courses on this if you'd like to learn more about process mapping: https://learn.nintex.com/series/process-mapping

The key is to identify where there are inefficiencies, name them and don't replicate them in your software solution

Step 3: Involve the people that will be using the software from the start

Begin with the understanding that it's not just the implementation team that should be involved in these crucial early meetings, the people on the ground who will be asked to use the tools should be at these as well. A common issue that arises is that the people who are actually required to use the new software aren't involved until too late in the process.

> *If people are not part of the early process, they won't feel a sense of ownership, they won't have buy in and they are the people that are going to ask the useful real world questions.*

Understanding these questions very early on is only going to benefit everybody and help you to identify the real problems.

The vendors you engage, they know their solutions, so it's important to avoid going in with too much of an end solution in mind and instead focus on a clear articulation of what you want their solution to achieve. But be open minded to how it could look, because it could actually be better than what you were expecting.

So keep in mind there's different requirements of what a software solution needs to achieve from a business perspective. And then there's the people that are actually going to be using it and their requirements.

Useful questions to ask:
Is it user friendly? Are they going to get a lot out of it? Yes, the outcome may suit the business, but actually getting your people to input the data to get the outcome is not going to happen if the solution isn't taking into consideration these end user needs.

Step 4: Communication is definitely a theme that runs through all of this – what are some of the essential things businesses need to get right before they even start rolling out the software?

Begin with the understanding that it's not just the implementation team that should be involved in these crucial early meetings, the people on the ground who will be asked to use the tools should be at these as well. A common issue that arises is that the people who are actually required to use the new software aren't involved until too late in the process.

High level:

- Implementing on 2-3 projects initially, to iron out the kinks before full roll out.
- Clear understanding of internal & external roles and responsibilities with the new software.
- Comms on what is being implemented, why, when and 'how this will affect you'.
- Comms on what this is not including, and what will remain the same.
- Clear program of how the roll out will work.
- A timeline for the implementation.

Internal communication

It may seem obvious, but there a few pieces businesses often miss when they communicate about a new piece of software. One email isn't going to cut it.

Of course it's important to let everyone in the organisation know ahead of time about a change in tools and process – the what. But what is this change going to mean? It's important to explain the why, how and when of the change.

Primarily what support will the team be getting, in terms of training and chances to have their questions answered.

Understand how changes to process will affect different cohorts across your business

Your people on site may have very different questions to someone in the front office. Connect with these people early on and address their concerns and questions. Also your IT team / person should be involved and aware, as we'll discuss in a moment.

If you have an intranet or internal blog an FAQ can be useful. Reach out to departments, put up posters in the lunchroom – you really can't overcommunicate here.

But apart from this as well, you want to talk about what won't be changing.

You've may have current processes and ways of doing things, that will not be changing and this can bring some comfort to those afraid of and resistant to change.

We always recommend to implement on two to three projects first.

There's going to be changes and refinements throughout the implementation process.

When you are evaluating your processes against the system you'll find that they inevitably change from when you have a system configured to out in the field.

It's best to use two to three projects and make it clear that these are experiments for testing the new software. Collecting refinements and understanding any variation in user types and feeding that back to the vendor will make wider roll out so much smoother and efficient.

Importantly if you know, someone that's not tech savvy, pair them up with someone that is, and then really understand what the pain points are.

Have you got the right equipment to do the job?

A super obvious consideration, but one that is often overlooked, is to ensure everyone is set up with decent computer equipment.

Giving a team member a laptop that is insanely old and super slow and then asking them to do inductions with contractors onsite is setting everyone up for failure. So check connection on sites, check the age of laptops, check devices.

If you're implementing an online system that's meant to be used out on site in real time, give your team the tools that enables them to get the benefits from the system, not just barely use it.

> *You're investing time and money into a new system, going to all that effort and not enabling users to get the full benefits is a waste.*

Stage 2 – Implementation

Step 1: You've made a purchase, now need to get everyone using it!
Don't have a team of learning and development people – what do you do?

High level:

- Understand that people generally operate from a modality of fear and will automatically revert to this mindset when change comes.
- They perceive their skillset is inadequate against software and see it as a threat not opportunity:
  - Sell it as to how it will specifically aid their position.
  - Don't move too quickly.
  - Lean into their expertise and their knowledge.
  - Listen, listen, listen.

So the financial manager has an amazing spreadsheet, comparing all of the options and has decided on the most cost effective solution. And that's the end of their involvement, the software is handed over to users. Hopefully based on what you've already read, it's no surprise that this sort of approach tends to be a disaster.

Besides all of our earlier points about involving the right people early on and communicating. There's often simply no infrastructure there to support the roll out and it fails.

Especially in smaller companies, the finance manager isn't the person who is then going to proceed with the implementation.

What does work across organisations big and small is to find the generalists in the organisation and really have them champion the system.

Find the generalists in the organisation and have them champion the system."

For example, this might be the operations manager or the office manager. Whoever they are, they need to be first and foremost excellent communicators and influencers. Just because someone may be tech savvy, or have promoted the software from the start doesn't mean they're necessarily going to be able to champion the system. So make sure you choose your champions wisely.

It is a process of really adopting change management principles: Share the bigger picture, and lead by example.

The leaders of the organisation have to be the first people to use it, and then involve everyone that is affected by this implementation and build a plan around it. It's also to share with everyone an idea of what the metrics are for success.

How you will all know when it's working? You need to make sure that you can measure the results and communicate when that result is reached, because that makes people feel good about it.

But as with everything that you do, not everyone is going to love it at first, there is going to be huge resistance.

So work closely with the early adopters, because they're what's going to help the implementation actually take off and be more recognised within an organisation, and then work with the lot, the late majority.

So your early majority people are really going to be the ones who will make this a success and they're the ones who you're going to empower, empower those people which will then have their sphere of influence to make sure that the balance will also come into it.

So you give the power to the people. You give it to an individual, not a department.

Step 2: Leverage Your Vendor Partner

It's important to leverage your vendor partner because, more than likely, they've been at the coalface of a lot of successful implementations.

Ensure your internal champions have access to and work collaboratively with your vendor partner as part of the change management process. The people (or person) implementing this within the organisation. They're not necessarily the expert, they might be keen and ready to go but they can also benefit from the years of experience within the actual companies who are rolling this out for you.

The help and reference materials that you get from your software provider is often another element that is overlooked. Some companies will supply a 50 page manual that goes into great detail of every feature of the software, or send you to a website with hundreds of pages of material.

Let's be honest, there's not too many people that will read a 50 page manual, let along poke around your help site for hours.

You're well within your rights to just ask you ask your software partner to provide a one pager, or some more focused onboarding materials.

Step 3: Change Management

First of all, it's really important to understand that 90% of people are operating with a 'fear modality', only 10% of people are actually able to see the big picture immediately.

So if 90% of the people are operating out of fear, they're going to automatically revert to a fearful mindset when any sort of change comes up. Understanding that fear based response is very important.

And what are they typically afraid of? Becoming irrelevant, or becoming redundant.

But it's nothing to do with the inadequacy of their skill set or their job, software is rarely actually a threat. It's an opportunity, because these systems then free up the individual to actually do their job, not administration. Working with people collaboratively, not pushing paper. Software can give people more scope to become better at their jobs.

But it's important to sell how it will specifically aid their position. Only then will you start to coax them out of their fear mode and into the vision mode.

People are fundamentally going to focus on "what's in it for me", and if they don't know what's in it for them, they are not going to do it. This is why getting folks involved early on is so key.

When communicating to different cohorts about the new system actually itemise exactly what's in it for them

A lot of this resistance also comes around when things happen too quickly. So going back to that communication plan, you want to ensure you reach people, so they're not constantly being scared because things are changing too quickly, and they can't see forward enough.

As we mentioned earlier, it is about leaning into their experience early on , they're at the coal face, they know their pain points better than anyone else.

And when people feel that they're not being listened to, it causes frustration. So listening is absolutely key instead of trying to override what they're feeling, which is not really going to be rational, but also listening to the logical side because they're going to help iron out a lot of the resistance and the bumps in the road to make sure that that your rollout is successful.

Step4: Timing is Key

Timing the onboarding and training of your software is very important. You want to make sure the gap between when you onboard staff and when they are ready to start using it is not two, three months down the line.

Again that is where you can work together with your with your software provider and ensure you're scheduling the onboarding session at a time that suits you and when the most people are going to start using the software.

We understand, you've bought software, you're excited. So you might want to do that onboarding session straight away. But in reality, if your team is not going to be using it for a month, or even six months down the line, (which is often the case), people will forget and it's not going to have that momentum in your business.

Another error to avoid if possible is rolling out new software on a project when it is halfway through, at its peak and everyone is super busy.

Stage 3 – Health Check

Put baselines in place that help measure the efficacy of the software. Has it delivered on its promises? If not how do you make sure that's not because your process isn't calibrated correctly.

High level:

- You need to have measurable yardsticks about how you will judge the software. For example; "Within 3 months I want to have the software live on 3 projects and have 10 competent users." The more precise you are with this, the easier it will be to measure the efficacy of the software.
- Have an open dialogue and ask lots of questions to your account manager/onboarding specialist in the early stages of implementation. This will give you a clear picture of what the software can and can't do, and will avoid any confusion around issues that may occur within your own processes that your new software isn't actually designed to fix.

It's important to set clearly definable yardsticks that allow you to judge how successful your software is at addressing the problems you identified right at the start and how successful your roll out has been in order to assess this properly.

For example, you might say within three months of implementing the software you'd like to have the software live on three projects and have 10 competent users within the business. These are the defined metrics. And as we mentioned earlier it's important to make everyone on the project aware of these metrics early on.

It can be as simple as at the end of three months, running a bit of an internal survey of your team and seeing how they feel about using the software. If you've got 10 out of 10 users who like it and it's live on three projects, you've fulfilled your criteria.

The more precise you are with these metrics from the outset, the easier it is to measure how efficient the software has been for you. And be aware it will take time.

No software will improve the efficiency of your business overnight. It's going to be a gradual thing no matter how good the solution is. Getting the team up to speed and confident, then refining could take months. It's a good idea to manage expectations around this fact and be very precise about your success metrics – especially with your vendor.

No software will improve the efficiency of your business overnight. It's going to be a gradual thing no matter how good the solution is.

If you come back after three months and you're not quite live on three projects, it's very easy to ask, why not? Whereas if you just say, I want to be happy with the software after three months, what does that mean?

This is where you can have an open dialogue with your vendor's onboarding specialists or account manager, after all they want you to be successful.

The worst thing that you can occur is when you have problems that you want to solve using the software, that the software you've bought, actually isn't designed to solve.
And because that only creates frustration, it's so important to have a very clear dialogue from the beginning with the onboarding specialists from your software company. They'll be able to advise you straightaway on areas that the software is absolutely going to be able to help your business, but also areas where you might have to look at outside of the particular product.

Also, even if a vendor can't solve those particular problems now, that's something that can be feedback and potentially added to a product.

So, the communication and relationship with your vendor and the account manager, implementation specialist or customer success manager is very important, especially in the first three to six months.

But equally important, is keeping those lines of communication going long term. because scenarios may emerge that you're not be able to handle on your own.

Implementation Checklist on a Page

- □ System capabilities vs your requirements – getting these aligned and clear
- □ Does everyone know your current processes or will they need to be trained along with the new software?
- □ What is your current technology setup i.e. connection on site, age of laptops, access to tablets and involve your IT provider
- □ Nominate a project to implement on first, to get the process correct, before rolling out
- □ Nominate people that will be implementing the software, to be part of the decision making process
- □ Focus on needed requirements, not nice to haves.
- □ Establish yardsticks about how you will judge the software.
- □ Map existing processes before you transfer over to software to gain clarity and eliminate inefficiencies.
- □ Communicate understanding of internal & external roles and responsibilities with the new software.
- □ Comms on what is being implemented, why, when and 'how this will affect staff.
- □ Comms on what this is not including, and what will remain the same.
- □ Clear program of how the roll out will work.
- □ Establish optimal roll-out time and publish a timeline for the implementation.
- □ Communicate with your external partners, share the above information.
- □ Share training materials and schedule follow up training with internal and external parties as needed
- □ Share your plan with software vendor and get their feedback, they may be able to provide assistance
- □ Once the software has been in place for the agreed period 'health check' the system with staff – is it working as promised, has the roll-out met your yardsticks?

Authorship Note

This e book is based on the webinar "A practical guide to construction software implementation" held on the 4th of June 2020. We'd like to thank our panel for their contribution:

Sarah Roe - Product Manager, HammerTech
Angus McIntyre – Customer Success Manager, Payapps
Elinor Moshe - Founder of the Construction Coach
Nick Opasinis – Senior Business Development Manager, Payapps

PAYAPPS
CONSTRUCTION PAYMENT APPLICATIONS
A ZUUSE SOLUTION

About Payapps

Payapps bridges the gap between multiple organisations and stakeholders, by providing an easy-to-use, real-time status of progress claims. Developed specifically for the construction industry, it allows everyone to collaborate on a single platform, eliminating the need for endless spreadsheet reconciliations, email trails and phone calls.

Pay us a visit at payapps.com

ABOUT US

HAMMERTECH *is a cloud-based, mobile performance and productivity platform, ensuring operational efficiency, quality, & safety. If you're looking to promote safety & quality on your jobsite, it's time to focus on operations. HammerTech is the ONLY operational platform to cover pre construction operations, daily operations, safety & quality. Manage online registrations & inductions, employer management, equipment induction & monitoring, permit management, SWMs and SDS management, audits, compliance, observations, defect lists and more within one mobile, configurable platform. Consolidate your apps, and go all in one.*

Learn More:

1. See inside the software and get a personalised tour: https://go.hammertechglobal.com/demorequest
2. Stay up to date with the latest news and follow us on LinkedIn: https://www.linkedin.com/company/hammer-technologies-pty-ltd/?viewAsMember=true
3. Download this software checklist to see how your current solutions match up to the HammerTech platform: https://cdn2.hubspot.net/hubfs/5108829/HT_Q+S-Checklist_0219.pdf