Managing Contracts and People

Introduction:

When the project process begins, few things are as important as the details involved in the relevant contracts. In the construction industry, many contracts combine to provide parameters and specifics that set the tone for the entire process. Once executed, timeliness and tracking of requirements, requests and promises are an important priority.

If information hasn't been entered into a managed, standard system, the paper or digital contracts can get lost in the shuffle, or lose priority status as they are routed from team member to team member—a recipe for poor performance, unanswered questions and bad reputations.

What is Contract Management?

Contract management is an intricate oversight process that follows contracts from pre-award to completion, including execution, vendor selection, issue detection and control, tracking and processing. When implemented properly, contract management processes ensure that budgets and abilities are in alignment with project objectives.

The best contract management flows seamlessly through the organization and integrates with project management and control, always involving the team members for input and outcomes, and carefully monitoring contractors for performance and deadlines.

While contract management can be beneficial in any industry, it is vital for industrial and construction projects. In fact, contracts in these industries ensure that operations and financial goals are met and risk is reduced. Supervision, quality control and the ability to manage numerous professionals at a time can all be handled by a contract management program specifically implemented for these types of projects.

The Stages of Contract Management

Contract management is not solely about creating agreements and getting them approved. It includes a series of stages that follow the process through to a successful conclusion. Any missed steps can cause delays and mistakes down the line. Here's an outline of five fundamental areas of importance:

1. **Create:** The contract management system must have the ability to incorporate standardized procedures with details specific to the goals of the organization. First steps include identifying the type of contract and who will be responsible for each task. The planning process should consider company resources, objectives and team member strengths and weaknesses, while developing an overview of potential challenges and risks.
2. **Negotiate:** The contract should be written in a manner that reflects the organizational needs and values, helping to establish trust between the two parties. Once the initial contract has been designed, negotiation is the obvious next step. Line items can be discussed, changed or removed, as part of the negotiation process.

3. **Approve:** Approval usually includes multiple sign-offs. Numerous managers, departments and even contractors, may have to sign off on the specifics before the final deal is made.
4. **Finalize:** The contract signing process between enterprises is the final step before getting the project underway. Obtaining signatures from numerous parties and entities quickly—even when distance is a factor—is crucial to avoiding postponements to the process.
5. **Manage:** Once the project begins, changes can still occur. Revisions need to be carefully managed and communicated to the appropriate parties. Deadlines, audits, revenue, and expenditures need to be tracked, completed and shared with the rest of the team.

A standardized program that is customizable to an organization's specific needs helps a contract run smoothly. When these five areas are carefully managed, the odds of a successful relationship and a closed deal go up exponentially.

Why is Contract Management Important?

A contract provides parameters for key aspects of a project including business strategy and relationships. Competing in today's global marketplace means team members can be located all over the world, adding challenges concerning time-zone, cultural needs and understanding.

Contracts include payment terms, negotiations, workflow, service expectations, and compliance obligations. Contract management helps reduce risk by ensuring compliance when necessary, as well as providing monitoring and tracking to prove it. It also augments the bottom line by comparing prices and reducing spending. Real-time visibility of all aspects of a contract over time enables management to make data-driven decisions and necessary course corrections before it's too late.

When a contract is initiated, it should reflect goals, timelines, budgets, resources, risks, regulations, and specifications. Each phase of the process requires specific elements, purpose and management in order to proceed to the next step.

Technicians, engineers, and other skilled professionals must be carefully chosen to complete the contract and execute the project. Technology provides an active thread connecting all aspects of the project, helping to fill in the holes during revisions, and ensure communication with the right team members, at the right time.

Many construction companies are faced with re-thinking how business processes like productivity, performance, labor, and inventory are affecting growth and profitability. Communication, tracking, managing revisions and a variety of other tasks can add to an already complicated process. Good contract management and operational efficiency go hand-in-hand.

Contract Management Challenges

Tracking and managing the array of moving parts of a contract can present a variety of challenges. In the engineering and construction industry, projects are often large and

complex, involving joint ventures, numerous sub-contractors and complications related to global execution.

The processes of contract management can present serious challenges, especially when handled manually. Specific challenges include:

- **Execution:** Without proper management, files are easily lost, and final approvals and signatures take much longer than necessary.
- **Tracking:** It can be difficult to track and audit contracts once they have been signed and are passed on to other staff who may not understand the details that must be monitored. Budget information, e-mails, information from important meetings, and changes are stuck in transit before being addressed. Time is money, and project delays decrease profitability.
- **Revisions:** Managing change before a contract is approved and after it has been executed is equally important. Larger projects frequently include national or global teams and create obstacles to deadlines and cohesiveness. Updates to original documentation can take days, or even weeks. What's more, if teams are working from different versions, the wrong decisions could be implemented. Any of those scenarios could lead to increased risk, missed deadlines, mistakes and even litigation.
- **Compliance:** Details laid out in the contract are non-negotiable obligations. From a legal standpoint there may be regulatory or compliance issues that must be followed to the letter. Failure to comply can have serious legal and/or financial consequences.

Benefits of Contract Management

Contract management improves communication, response time, goal alignment, transparency and accountability. These and other project performance metrics can be tracked and improved with good contract management:

- **Efficiency:** Contract management streamlines adherence to the contract and can lower business costs. All necessary documents can be found and accounted for in one place, offering increased transparency for team members from different departments, as well as contractors working offsite. The addition of automation makes documents and changes more accessible, which can, in turn, reduce response time to changes, additions or challenges. Positive experiences relating to contractual agreements may lock in better raw material pricing and availability as well as service costs down the line.
- **Risk reduction:** Important business objectives and goals are identified when a contract is written. A good contract management process sets expectations around those priorities and ensures commitments in the contract are met. The benefit works all around, holding organizations to performance clauses, and providing vendor compliance. Regulations are in a continual state of change, and staying on top of them means constant oversight to ensure your projects stay in compliance. Tracking assets, obligations and provisions is a must to mitigate the risk of financial loss. The tracking element provided by a contract management tool constantly gauges performance and compliance in real-time.
- **Relationship building:** A positive contract experience creates lasting business partners with vendors and subcontractors. Particularly in the construction industry, finding good help is paramount for future projects.

- **Tracking and documentation:** Contract management tools organize and centralize documents and processes making business insight and analytics more accessible. Standardized reporting and record-keeping results in accuracy and visibility, which, in turn, creates actionable insights. When contracts are managed with transparency, projects are less likely to get stuck because of an unknown challenge or compliance issues.

Types of Contracts:

The types of contracts in software project management can include fixed price, firm fixed price, fixed price incentive fee, fixed price with economic price adjustments, purchase orders, cost reimbursable, cost plus fixed fee, cost plus incentive fee, cost plus award fee, cost plus percentage of cost, time and materials, and unit price contracts.3 min read

1. Fixed Price Contracts
2. Purchase Orders
3. Cost Reimbursable Contract
4. Unit Price Contract
5. Time and Materials Contract

The types of contracts in software project management can include fixed price, firm fixed price, fixed price incentive fee, fixed price with economic price adjustments, purchase orders, cost reimbursable, cost plus fixed fee, cost plus incentive fee, cost plus award fee, cost plus percentage of cost, time and materials, and unit price contracts.

Fixed Price Contracts

With fixed price contracts, also known as lump sum contracts, the buyer and service provider agree on a fixed price for the services in question. This type of contract is low-risk for the buyer, but high-risk for the seller since the time and costs of the project could exceed the fixed price. For this reason, a fixed price contract should include a detailed scope of work that clearly outlines what the buyer can expect for the agreed-upon price. When the contract is signed, the seller must complete the task or deliver the goods as agreed or risk being in breach of contract. Types of fixed price contracts include the following:

- **Firm Fixed Price Contracts:** This type of fixed price contract is typically used in government and partial government projects where the scope is defined in detail. This makes it easy to create a request for proposals and to compare the bids you receive. The downside for this contract is that deviating from the defined scope can be expensive.
- **Fixed Price Incentive Fee Contracts:** With this type of fixed price contract, the buyer also offers a performance-based incentive as an extra payment to the seller. Performance can be measured for this purpose by various metrics, including time, cost, or performance.
- **Fixed Price Award Fee Contracts:** As with the fixed price incentive fee, this type of contract offers a bonus for exceeding a specific performance metric. For example, if the seller delivers the product early, he or she could be eligible for a bonus equal to 10 percent of the total contract.

- **Fixed Price With Economic Price Adjustment:** With this type of contract, although the price is fixed, it can be readjusted with fluctuations in the market.

Fixed price contracts are commonly used on a deliverables basis for outsourcing and turnkey procurement.

Purchase Orders

A purchase order is a specific type of contract that is used only to purchase goods and commodities.

Cost Reimbursable Contract

When the scope of a project is unclear or subject to change, you should consider a cost reimbursable contract. This document, sometimes called cost disbursable, is also useful when the risk of a specific project is high. The seller provides work for a fixed time period or project, then increases the bill to create profit after finishing the work.

The amount of profit in this type of contract is often based on performance metrics detailed in the document itself. The downside of this type of contract lies with the buyer, who carries the risk for this type of contract since he or she pays all costs. The full cost of the contract won't be defined until the work is done. For this reason, few businesses opt to use a cost reimbursable contract.

Types of cost reimbursable contracts include the following:

- **Cost Plus Percentage of Costs/Cost Plus Fee:** With this variation, the seller receives a defined percentage of the total cost of the project upon completion. This is also an arrangement that mainly benefits the seller.
- **Cost Plus Fixed Fee:** The seller receives costs incurred plus an additional flat fee that is fixed in the contract. This amount is received when the contract is fulfilled regardless of performance.
- **Cost Plus Incentive Fee:** This is a performance-based fee paid on top of actual costs. It is a flat amount rather than a percentage.
- **Cost Plus Award Fee:** Similar to a cost plus incentive fee, this contract provides an award on top of the costs incurred.

Unit Price Contract

This type of contract, also called an hourly rate contract, combines elements of fixed price and cost contracts. A unit price contract pays a specified hourly rate for every hour spent on the project. It is commonly used by freelancer workers.

Time and Materials Contract

This contract is used when labor is the main deliverable and typically provides the seller an hourly rate.

If you need help with software project management contracts, you can post your job on UpCounsel's marketplace. UpCounsel accepts only the top 5 percent of lawyers to its site. Lawyers on UpCounsel come from law schools such as Harvard Law and Yale Law and average

14 years of legal experience, including work with or on behalf of companies like Google, Menlo Ventures, and Airbnb.

Stages in Contract:

Lifecycle. Process. Whatever you call it, effective contract management doesn't only involve developing agreements and getting them signed – it's a series of actions that guide you from the earliest stages of developing holistic processes for handling each and every company agreement, through to the steps to seeing contracts through to their conclusion.

Having a clear understanding of what happens at each stage is an important way to ensure your contract management processes meet all of the requirements and objectives to deliver optimal results.

Here are the seven essential stages of contract management.

1. Planning stage

Before you can implement a process, it's important to develop a system that will best suit your company's needs and resources. To keep things streamlined and organized, it's also important to develop contract management processes that can be implemented company-wide.

Your contract management strategy is a flexible roadmap consisting of processes that account for all types of company agreements, from standard employment contracts to the paperwork from highly specific and complex deals. The first step to developing your strategy is to determine your needs, including answering the following:

- What types of contracts do you have to manage and in what volume?
- Are there standard agreements you use again and again? What needs to be included in these?
- Who is responsible for each stage of contract management and what do they need to perform their job?
- What common problems have occurred in the past, or what issues might arise during the management of a typical contract?
- What resources are required to implement your contract strategy?

Understanding the remaining stages of contract management will help to inform your processes.

2. Implementation stage

Once you have outlined your contract management processes, you will need to implement your plan before you can start using it. This includes deploying contract management software to help you to execute contract-related tasks, as well as migrating your contracts to a centralized repository.

A crucial part of your implementation plan is onboarding – making sure everyone involved understands your vision and objectives for contract management and is comfortable with the tools they will be using.

3. Pre-contract stage

Now that you have your contract management foundation set up, you can begin to implement it for new contracts. That means developing new contracts or implementing boilerplate agreements for more standard situations. The key challenge of this stage of contract management is developing a specific document that will deliver what you need and reduce your risks.

For standard situations, this stage may be as simple as finding the right contract type, entering the relevant information and perhaps making a few tweaks. More unusual or complex contracting scenarios may require the development of a whole new document. Developing a contract from scratch can be made easier by looking at other agreements that might be applicable and adapting those terms. Don't forget to carry over any important requirements such as compliance obligations or branding standards. Once you have agreed on the terms and developed your contract, e-signatures can keep things moving.

4. Handover stage

It's common – especially in larger companies – that the individuals involved in executing a contract are not the same as those who negotiated it. Thus, in order to ensure the contract is fulfilled as expected, it's important to ensure a smooth handover. Rather than assuming stakeholders have everything they need, it's useful to spend some time walking through all of the contract details and confirming roles, responsibilities and milestones.

5. Contract stage

The contract stage is when all of the goals of your contracts come to life – if you manage them properly. And much of the contract management work you've performed thus far is setting you up to do just that.

But the contract stage doesn't manage itself – it's here where you must play close attention to all of the terms laid out within your agreement and perform regular monitoring to make sure everything is happening as it should. It's useful to have a plan for doing so, with a clear sense of key milestones and performance metrics that will let you confirm everything is on track – or provide an early warning system if any problems arise.

6. Pre-renewal stage

Nothing lives forever – not even your contracts. But there are several ways your agreements may come to an end: one-off agreements may wind down to a natural conclusion, you may renew an agreement, or choose to terminate it. Often there are specific terms – and even possibly penalties or default actions, should you fail to do anything – that can affect the outcome, which is why it's important to start thinking about the end of your contract in a

proactive and timely manner. Now is the time to evaluate how your contract performed and decide whether you want to renew and/or make any changes. Make sure all stakeholders are aware of termination and renewal dates and that you have enough time to consider all the information before you get locked into any decisions.

7. Post-contract stage

Once a contract ends, there is still some housekeeping to do to ensure that everything is wrapped up properly. This includes ensuring termination conditions have been met, issuing or paying final invoices, and archiving your contract. It's also useful to perform a contract post-mortem, which can provide valuable information and learnings that can improve the results of future contracts.

Placement:

In a textbook such as this, it is not possible to describe all the necessary content of contracts for IT goods or services. It is possible, however to outline some of the major areas of concern.

Definitions

The terminology used in the contract document may need to be defined, for example, who is meant by the words 'client' and 'supplier'.

Form of agreement

For example, is it a contact of sale, a lease, or a licence? Also, can the subject of the contract, such as a licence to use a software application, be transferred to another party?

Goods and services to be supplied

Equipment and software to be supplied This includes an actual list of the individual pieces of equipment to be delivered, complete with the specific model numbers.

Services to be provided This covers such things as:

• documentation;

• installation;

• conversion of existing files;

• maintenance agreements;

• transitional insurance arrangements.

Ownership of the software

Who has ownership of the software? There are two key issues here: firstly, whether the customer can sell the software to others and, secondly, whether the supplier can sell the software to others. Where off-the-shelf software is concerned, the supplier often simply grants a license for you to use the software. Where the software is being written specially for a customer, then that customer will normally wish to ensure exclusive use of the software - they may object to software which they hoped would give them a competitive edge being sold to their rivals. They could do this by acquiring the copyright to the software outright or by specifying that they should have exclusive use of the software. This would need to be written into the contract. Where a core system has been customized by a supplier, then there is less scope for the customer to insist on exclusive use.

Where software is written by an employee as part of a contract of employment, it is assumed that the copyright belongs to the employer. Where the customer organization has contracted an external supplier to write software, the contract needs to make clear who is going to retain the copyright - it cannot, in this case, be automatically assumed it is the customer. The customer might have decided to take over responsibility for maintenance and further development once the software is delivered and in this case will need access to the source code. In other cases, where the customer does not have an adequate in-house maintenance function, the supplier can retain the source code, and the customer will have to approach the supplier for any further changes. There are many potential dangers with this, not the least being that the supplier could go out of business. An escrow agreement can be included in the contract so that up-to-date copies of the source code are deposited with a third party. In the United Kingdom, the National Computing Centre provide an escrow service.

Environment

Where physical equipment is to be installed, the demarcation line between the supplier's and customer's responsibilities with regard to such matters as accommodation and electrical supply needs to be specified. Where software is being supplied, the compatibility of the software with the existing hardware and operating system platforms would need to be confirmed.

Customer commitments

Even when work is carried out by external contractors, a development project still needs the participation of the customer. The customer will have to provide accommodation for the suppliers and perhaps other facilities such as telephone lines.

Acceptance procedures

Good practice would be to accept a delivered system only after it has undergone user acceptance tests. This part of the contract would specify such details as the time that the customer will have to conduct the tests, deliverables upon which the acceptance tests depend and the procedure for signing off the testing as completed.

Standards

This covers the standards with which the goods and services should comply. For example, a customer can require the supplier to conform to the ISO 12207 standard relating to the software life cycle and its documentation (or, more likely, a customized sub-set of the standard). Within the European Union, government customers with contracts for projects above a certain threshold value must, by law, ensure that the work conforms to certain standards.

Some customers find that specially written or modified software is not thoroughly tested by the supplier before delivery. Some suppliers seem to think that it is cheaper to get the customer to do the testing for them!

Project and quality management

The arrangements for the management of the project must be agreed. Among these would be frequency and nature of progress meetings and the progress information to be supplied to the customer. The contract could require that appropriate ISO 9000-series standards be followed. The ISO 12207 standard provides for the customer to have access to quality documentation generated internally by the supplier, so that the customer can ensure that there is adherence to standards.

Timetable

This provides a schedule of when the key parts of the project should be completed. This timetable will commit both the supplier and the customer. For example, the supplier might be able to install the software on the agreed date only if the customer makes the hardware platform available at that point.

Price and payment method

Obviously the price is very important! What also needs to be agreed is when the payments are to be made. The supplier's desire to be able to meet costs as they are incurred needs to be balanced by the customer's requirement to ensure that goods and services are satisfactory before parting with their money.

Miscellaneous legal requirements

This is the legal small print. Contracts often have clauses that deal with such matters the legal jurisdiction that will apply to the contract, what conditions would apply to the sub-contracting of the work, liability for damage to third parties, and liquidated damages. Liquidated damages are estimates of the financial losses that the customer would suffer if the supplier were to fall short of their obligations. It is worth noting that under English law, the penalties laid down in penalty clauses must reflect the actual losses the customer would suffer and cannot be unrealistic and merely punitive. Even this limitation will not be enough in some cases as far as the supplier is concerned. As computer systems assume increasingly critical roles in many organizations and in safety-critical systems can even be life-threatening in the case of malfunction, the possible consequential damage could be very great. Suppliers will not unnaturally try to limit this kind of liability. The courts (in England and Wales) have tended to look critically at such attempts at limiting liability, so that suppliers will, in the case of major contracts, take out insurance to cover such liabilities.

If there is a dispute, resorting to litigation, while being lucrative to the lawyers involved, is both time-consuming and expensive. An alternative is to agree that disputes be settled by arbitration. This requires that any dispute be referred to an expert third party whose decision as to the facts of the case is binding. Even this procedure is seldom quick and inexpensive and another option is alternative dispute

Typical Terms of Contract:

In a textbook such as this, it is not possible to describe all the necessary content of contracts for IT goods or services. It is possible, however to outline some of the major areas of concern.

Definitions

The terminology used in the contract document may need to be defined, for example, who is meant by the words 'client' and 'supplier'.

Form of agreement

For example, is it a contact of sale, a lease, or a licence? Also, can the subject of the contract, such as a licence to use a software application, be transferred to another party?

Goods and services to be supplied

Equipment and software to be supplied This includes an actual list of the individual pieces of equipment to be delivered, complete with the specific model numbers.

Services to be provided This covers such things as:

• documentation;

• installation;

• conversion of existing files;

• maintenance agreements;

• transitional insurance arrangements.

Ownership of the software

Who has ownership of the software? There are two key issues here: firstly, whether the customer can sell the software to others and, secondly, whether the supplier can sell the software to others. Where off-the-shelf software is concerned, the supplier often simply grants a license for you to use the software. Where the software is being written specially for a customer, then that customer will normally wish to ensure exclusive use of the software - they may object to software which they hoped would give them a competitive edge being sold to their rivals. They could do this by acquiring the copyright to the software outright or by specifying that they should have exclusive use of the software. This would need to be written into the contract. Where a core system has been customized by a supplier, then there is less scope for the customer to insist on exclusive use.

Where software is written by an employee as part of a contract of employment, it is assumed that the copyright belongs to the employer. Where the customer organization has contracted an external supplier to write software, the contract needs to make clear who is going to retain the copyright - it cannot, in this case, be automatically assumed it is the customer. The customer might have decided to take over responsibility for maintenance and further development once the software is delivered and in this case will need access to the source code. In other cases, where the customer does not have an adequate in-house maintenance function, the supplier can retain the source code, and the customer will have to approach the supplier for any further changes. There are many potential dangers with this, not the least being that the supplier could go out of business. An escrow agreement can be included in the contract so that up-to-date copies of the source code are deposited with a third party. In the United Kingdom, the National Computing Centre provide an escrow service.

Environment

Where physical equipment is to be installed, the demarcation line between the supplier's and customer's responsibilities with regard to such matters as accommodation and electrical supply needs to be specified. Where software is being supplied, the compatibility of the software with the existing hardware and operating system platforms would need to be confirmed.

Customer commitments

Even when work is carried out by external contractors, a development project still needs the participation of the customer. The customer will have to provide accommodation for the suppliers and perhaps other facilities such as telephone lines.

Acceptance procedures

Good practice would be to accept a delivered system only after it has undergone user acceptance tests. This part of the contract would specify such details as the time that the customer will have to conduct the tests, deliverables upon which the acceptance tests depend and the procedure for signing off the testing as completed.

Standards

This covers the standards with which the goods and services should comply. For example, a customer can require the supplier to conform to the ISO 12207 standard relating to the software life cycle and its documentation (or, more likely, a customized sub-set of the standard). Within the European Union, government customers with contracts for projects above a certain threshold value must, by law, ensure that the work conforms to certain standards.

Some customers find that specially written or modified software is not thoroughly tested by the supplier before delivery. Some suppliers seem to think that it is cheaper to get the customer to do the testing for them!

Project and quality management

The arrangements for the management of the project must be agreed. Among these would be frequency and nature of progress meetings and the progress information to be supplied to the customer. The contract could require that appropriate ISO 9000-series standards be followed. The ISO 12207 standard provides for the customer to have access to quality documentation generated internally by the supplier, so that the customer can ensure that there is adherence to standards.

Timetable

This provides a schedule of when the key parts of the project should be completed. This timetable will commit both the supplier and the customer. For example, the supplier might be able to install the software on the agreed date only if the customer makes the hardware platform available at that point.

Price and payment method

Obviously the price is very important! What also needs to be agreed is when the payments are to be made. The supplier's desire to be able to meet costs as they are incurred needs to be balanced by the customer's requirement to ensure that goods and services are satisfactory before parting with their money.

Miscellaneous legal requirements

This is the legal small print. Contracts often have clauses that deal with such matters the legal jurisdiction that will apply to the contract, what conditions would apply to the sub-contracting of the work, liability for damage to third parties, and liquidated damages. Liquidated damages are estimates of the financial losses that the customer would suffer if the supplier were to fall short of their obligations. It is worth noting that under English law, the penalties laid down in penalty clauses must reflect the actual losses the customer would suffer and cannot be unrealistic and merely punitive. Even this limitation will not be enough in some cases as far as the supplier is concerned. As computer systems assume increasingly critical roles in many organizations and in safety-critical systems can even be life-threatening in the case of malfunction, the possible consequential damage could be very great. Suppliers will not unnaturally try to limit this kind of liability. The courts (in England and Wales) have tended to look critically at such attempts at limiting liability, so that suppliers will, in the case of major contracts, take out insurance to cover such liabilities.

If there is a dispute, resorting to litigation, while being lucrative to the lawyers involved, is both time-consuming and expensive. An alternative is to agree that disputes be settled by arbitration. This requires that any dispute be referred to an expert third party whose decision as to the facts of the case is binding. Even this procedure is seldom quick and inexpensive and another option is alternative dispute

Contract Management:

Most companies place great importance on contracts when formulating an agreement, but tend to forget about them once they have been signed. However, the life cycle of a contract goes far beyond its signature. The contract should serve as a compass throughout the agreement. It should be consulted on an ongoing basis to ensure that each party meets the expectations of the Agreement. The terms of these investment contracts necessarily include

assumptions at the beginning of the process and are subject to change over the life of the project lifecycle. For this reason, CRMS must be able to capture a single case of agreed changes to the terms of the contract and take these changes into account in a verifiable and legally robust manner. With multiple decision-makers involved, the CRMS should also make accountability more transparent and enable faster decisions on variation proposals. In addition, the responsibilities between project managers and contract managers can be very different. The contract manager profession is a specialized professional field that requires knowledge of federal, state, and local regulations, as well as the U.S. legal system. However, it is difficult for a project manager to learn everything they need to know about contract management without extensive training and/or experience.

To really reap the benefits, contract management software would include the following: Within your contract management platform, your contracts are searchable by text, simplifying searching and allowing you to accurately navigate through a huge amount of data. Contracts have many steps and parts inside. This means that tracking and managing all parts of a contract can lead to many challenges, and even more so when there are hundreds of contracts for a project. Global projects and megaprojects often involve large and complex contracts involving joint ventures, many contractors and subcontractors. Contract management is a complex monitoring process that accompanies contracts from pre-award to closing, including execution, supplier selection, problem identification and control, follow-up and treatment. When implemented correctly, contract management processes ensure that budgets and capabilities are aligned with project objectives. Many companies hire a contract specialist or contract manager whose only job is to manage and maintain contracts. If you`re considering hiring a contract specialist, your company could most likely benefit from a contract management system. The best contract management software systems keep the contract visible as the anchor of the agreement and keep an eye on important deadlines and milestones throughout the project.

Contract Lifecycle Management MLCs help you get a retrospective of the agreement and determine why one agreement met your expectations and why another did not meet your expectations. You`ll also see where you can spend less money and recoup unpaid winnings. With this valuable insights, you can make these positive processes repeatable and integrate these processes into your platform. A contract contains parameters for key aspects of a project, including business strategy and relationships. Competition in today`s global market means that team members can be based anywhere in the world, creating challenges in terms of time zone, cultural needs, and understanding. A contract management software system eliminates the manual burden to create better contracts faster. The information you get about your contracts will translate and often inform other parts of your business. Most importantly, contract management software equips your business for growth. Typically, there are two or more legal entities or different parties involved in the project, usually in client/contractor or contractor/subcontractor relationships. These different parties must sign a contract before starting the implementation phase of a project. Steps (1) to (6) are not legally binding, but are increasingly difficult to ignore or reject.

Due to its duration and formal nature, arbitration and litigation are the most expensive ways to settle claims. Therefore, before signing the contract, it is worth carefully discussing with the other party which of the first 6 steps could be incorporated into the contract. When a contract is launched, it must reflect objectives, schedules, budgets, resources, risks, regulations and specifications. Each step of the process requires specific elements, purpose, and management to move on to the next step. Contract management software systems provide a unified platform for creating, modifying, tracking, and renewing contracts throughout their

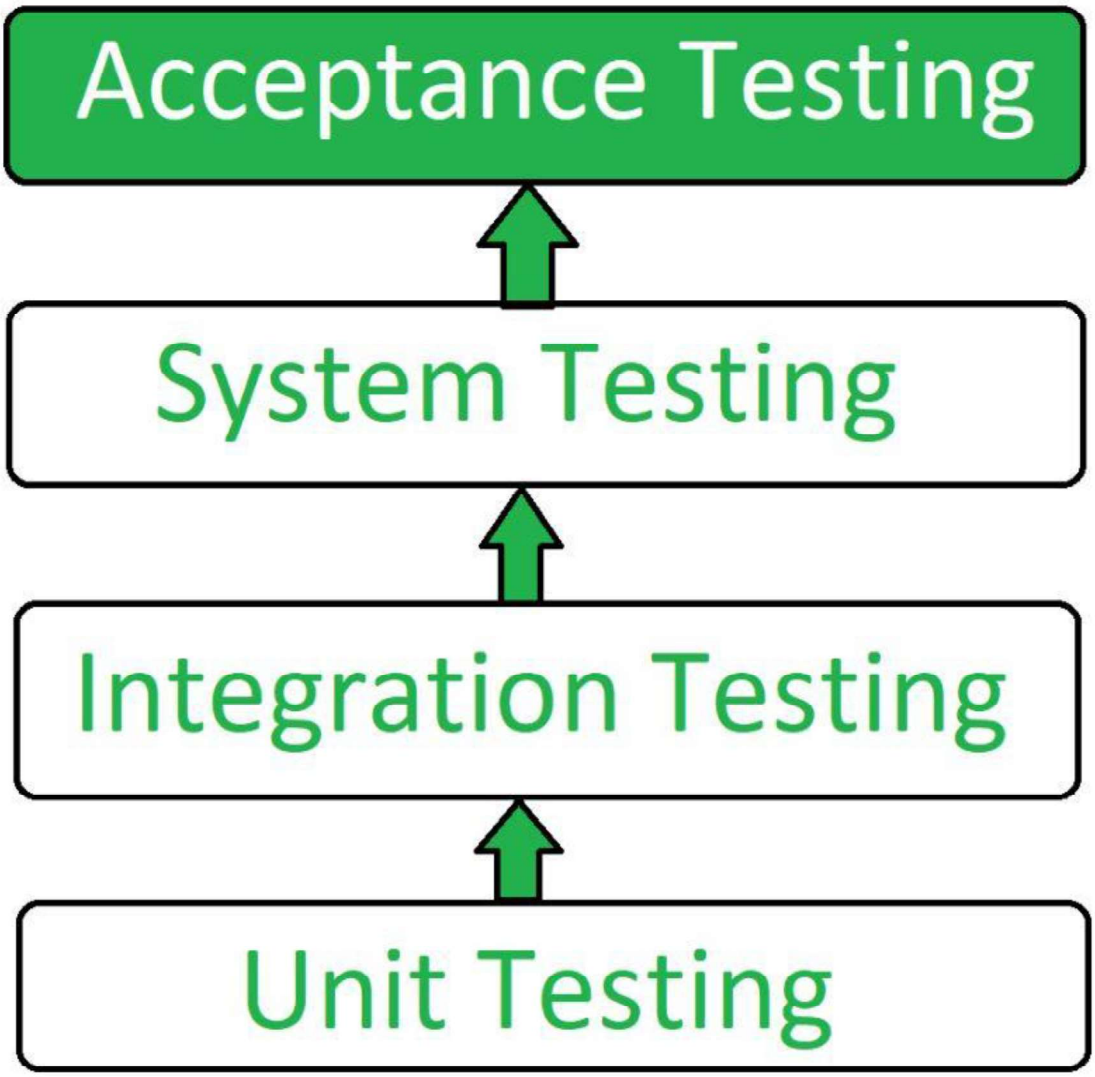
lifecycle. A CMS combines many features into one tool to provide countless benefits such as a central contract repository for information so your employees can easily find what they need. Models and other automations to reduce manual and repetitive tasks; and analytical reporting capabilities to help you analyze and optimize future contracts. Manually managing contract management processes can lead to critical challenges, including: Very large companies, such as. B investment projects (capex), involve several parties and high risks and uncertainties. They differ from traditional operating contracts in that they are subject to common deadlines in unique situations.

As the complexity of these unique projects increases, the relationships between the parties become more and more important. This requires contract management software or contract risk management software (CRMS) to become more dynamic and responsive. .

Zurück zur Übersicht

Acceptance:

Acceptance Testing is the last phase of software testing performed after System Testing and before making the system available for actual use.



Types of Acceptance Testing:

1. **User Acceptance Testing (UAT):** User acceptance testing is used to determine whether the product is working for the user correctly. Specific requirements which are quite often used by the customers are primarily picked for the testing purpose. This is also termed as *End-User* Testing.
2. **Business Acceptance Testing (BAT):** BAT is used to determine whether the product meets the business goals and purposes or not. BAT mainly focuses on business profits which are quite challenging due to the changing market conditions and new technologies so the current implementation may have to being changed which results in extra budgets.
3. **Contract Acceptance Testing (CAT):** CAT is a contract that specifies that once the product goes live, within a predetermined period, the acceptance test must be performed and it should pass all the acceptance use cases. Here is a contract termed a Service Level Agreement (SLA), which includes the terms where the payment will be made only if the Product services are in-line with all the requirements, which means the contract is fulfilled. Sometimes, this contract happens before the product goes live. There should be a well-defined contract in terms of the period of testing, areas of testing, conditions on issues encountered at later stages, payments, etc.

4. **Regulations Acceptance Testing (RAT):** RAT is used to determine whether the product violates the rules and regulations that are defined by the government of the country where it is being released. This may be unintentional but will impact negatively on the business. Generally, the product or application that is to be released in the market, has to go under RAT, as different countries or regions have different rules and regulations defined by its governing bodies. If any rules and regulations are violated for any country then that country or the specific region then the product will not be released in that country or region. If the product is released even though there is a violation then only the vendors of the product will be directly responsible.
5. **Operational Acceptance Testing (OAT):** OAT is used to determine the operational readiness of the product and is non-functional testing. It mainly includes testing of recovery, compatibility, maintainability, reliability, etc. OAT assures the stability of the product before it is released to production.
6. **Alpha Testing:** Alpha testing is used to determine the product in the development testing environment by a specialized testers team usually called alpha testers.
7. **Beta Testing:** Beta testing is used to assess the product by exposing it to the real end-users, usually called beta testers in their environment. Feedback is collected from the users and the defects are fixed. Also, this helps in enhancing the product to give a rich user experience.

Use of Acceptance Testing:

- To find the defects missed during the functional testing phase.
- How well the product is developed.
- A product is what actually the customers need.
- Feedback help in improving the product performance and user experience.
- Minimize or eliminate the issues arising from the production.

Advantages of Acceptance Testing :

- This testing helps the project team to know the further requirements from the users directly as it involves the users for testing.
- Automated test execution.
- It brings confidence and satisfaction to the clients as they are directly involved in the testing process.
- It is easier for the user to describe their requirement.
- It covers only the Black-Box testing process and hence the entire functionality of the product will be tested.

Disadvantages of Acceptance Testing :

- Users should have basic knowledge about the product or application.
- Sometimes, users don’t want to participate in the testing process.
- The feedback for the testing takes long time as it involves many users and the opinions may differ from one user to another user.
- Development team is not participated in this testing process.