active.collab

Project Management Methodologies and Frameworks

CONTENTS

PART I

Introduction to Project Management

What is project management

Project management is a set of activities which includes initiating, planning, executing, controlling and closing projects. It's a discipline that gives you principles, techniques, and tools to help you finish things on time and within budget.

The essential pillar of strength and support that holds up the entire project is the team of highly productive people who entirely devote themselves to it and make an all-out effort to get the job done.

Project management consists of a number of interlinking processes. A process is a series of actions performed by a team of people who work towards the same goal.

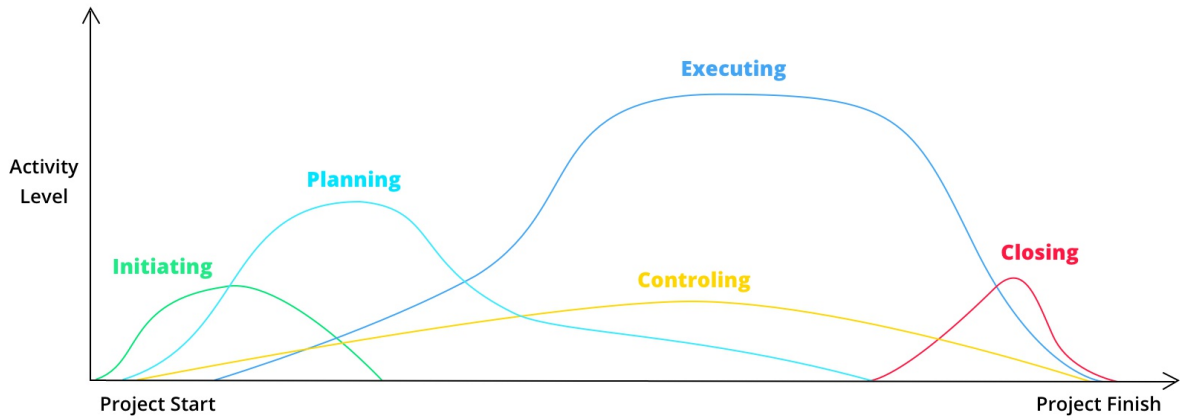
The 5 types of project management processes

According to the "Project Management Body of Knowledge" (aka. The PMBOK Guide, the standard guidebook for project managers around the world), there are 5 types of project management processes:

- **Initiating** - recognizing the beginning of the the project or a phase and that one phase can continue into the next one. Initiation process keeps the team focused on the business or halt the project if it fails to meet all the needs and preferences.

- **Planning** - creating a workable scheme that will include clearly defined activities, cost estimates, schedule development and resource planning.
- **Executing** - carrying out the processes which are followed by regular information distribution and team development.
- **Monitoring and controlling** - controlling the quality of project results, observing significant changes and making necessary adjustments to the project
- **Closing** - gathering all the necessary data to ensure that the projected is completed.

The processes do not necessarily follow a chronological pattern. Both in life and business, some things can get out of hand and are sometimes difficult to control. This is why processes usually overlap throughout different phases of the project and become ultimately dependent on one another.



How processes overlap on a project
Planning is the most important at the start of a project. Then, executing takes over. Controlling is important throughout the entire project.

Processes are linked by the outcomes they produce - the result of one process becomes an input to another. For example, the planning process provides the executing process with an early project plan. But the planning process is not over - it continues to play a role as it updates the plan as the project progresses.

Project planning is the most significant process as it should help you clearly layout all the details of the work to be done and predict possible hurdles you will need to overcome along the way.

You can’t accomplish any projects masterfully if you rely only on this framework of clearly defined processes. Project management is both a science and art. Each project is unique and tackling all the stages and issues that may arise is almost impossible without little bit of intuition and flexibility. You cannot know what steps you will be forced to take in order to achieve predictable results.

The key to project success is a good methodology combined with excellent intuitive skills.

What is project management and why it’s valuable

Projects have been practiced ever since the first human community was established. Plato writing the Republic, Michelangelo painting Sistine Chapel, Nikola Tesla’s designing the modern alternating current electricity system, Jorn Utzon building Sydney Opera House - all of them were the projects that moved boundaries of human understanding.

These huge and complex projects were endeavored by masterminds who invested considerable amount of time, energy and strong will to achieve their ultimate goal.

The incredible impact of modern technology has changed the face of business. As a result, project management became the essential part of any project, no matter how big or challenging it is.

People realized that using intuition and a mountain of paper documentation was neither efficient nor sufficient for running a successful business. Instead, they started viewing project management as a separate discipline, an invaluable methodology that will help them deliver the best possible results.

They needed a well-defined, well-thought and more disciplined process that will help all the ambitious entrepreneurs and their teams manage a project with breeze.

Project management is a huge benefit for everyone

Every ambitious entrepreneur has high hopes when starting a new project. Diving into a new project can be highly rewarding, but it also means taking on big risks. How many of us have successfully completed our projects without spending more money or investing more time in it?

Project management should help you:

- become more predictable

- save ample effort, time and cost
- resolve any issues more quickly and easily
- terminate “bad” projects
- encourage better communication and improve team’s productivity
- increase the chance to expand your services
- become more focused on the project development
- upscale the efficiency in delivering services
- become aware of the future risks and resolve them
- improve client collaboration
- create healthy working environment

Project management enables managers to run the project from the initial to the final phase in the most cost-effective and efficient way while keeping everyone in the loop.

Using technology to improve project management

A good communication flow is crucial for any successful business.

Today’s technology has done wonders when it comes to handling and distributing information worldwide.

There is a huge selection of integrated project management and communication tools that will help you control your activities more

efficiently and make your team more productive.

A project management tool helps virtual teams communicate better and be more productive. However, you shouldn't rely too much on technology because there are always some jobs that are not suited for it.

A number of business has skyrocketed over the last few decades and, according to research, as much as 55% of businesses understand the value project management delivers. But, not all of them practice it. Instead, many companies still rely on spreadsheets and interpersonal communication and, as a result, they fail to achieve the best possible results.

By using project management software, you will be able to handle information in a more affordable way and leverage your team's productivity. It's a true lifesaver for busy business owners looking for a way how to deliver their project in the most efficient way.

Long are the days when project management was viewed as merely a highly technical discipline. Today, project management is no longer a tedious time-consuming job as it used to be.

Due to the ever-growing array of business, managing resources and handling the team communication has become a real challenge. As a result, project management has become instrumental for managing any business successfully.

Project management life cycle

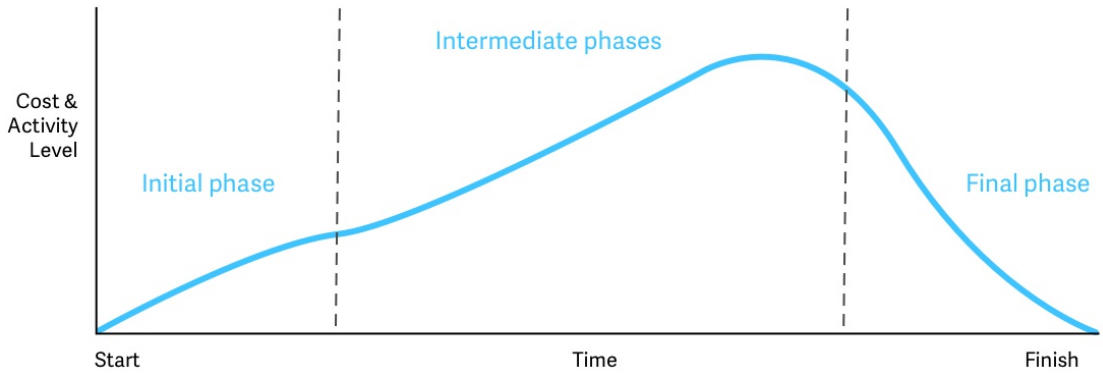
The project life cycle defines the beginning and the end of a project, including all the work in the middle.

No matter what methodology you use, it should include all the stages of an average project life cycle.

While many project life cycles have similar phases, only a few have the same number of them. The more complex and ambitious the project is, the more stages it will have before it's completed.

No matter how complex or simple the life cycle of the project is, all of them share the same pattern:

- Number of people, expenses, and activities are low in the beginning
- As the project progresses, numbers increase and finally drop when the project reaches the completion stage.



Project management life cycle phases

5 stages of project management life cycle

Stage 1: Initiating

In the initiation phase, you need to identify the business problem and, after carefully investigating all the options, come up with the most meaningful solution. Then you can initiate the project and start recruiting the team.

Initiating is the most critical stage in project management. Unless you meticulously define the project and build the solid foundation, you are running the risk of a project failure. It will create a domino effect, disrupting all the following stages as well as the final outcome.

Stage 2: Planning

After carefully defining the scope of the project, the next step is planning. Planning is the second most important phase of every project's life cycle because each project is unique and requires special approach.

The planning phase is an ongoing process which continues throughout the entire project. In its core, planning answers the questions:

- What should we focus on?
- What techniques will we use to accomplish our goal?
- When we are going to do it?

- How will we know if we successfully completed the project?

To successfully plan a project, you need to:

- Create a project plan
- Create a resource plan
- Create a financial plan
- Create a quality plan
- Create a risk plan
- Create an acceptance plan
- Create a communications plan
- Create a procurement plan
- Contract suppliers
- Review the whole plan

Stage 3: Executing

Executing is the part of the project management life cycle where you physically construct the deliverables and present them to the customer, who then decides whether he will accept them or not. It's usually the longest phase, how long depending on the project.

As the project manager, you control activities, resources, and costs while your team is performing the work identified in the plan. If you want to deliver on-time, you need to implement each activity

and introduce your team with the process as clearly as possible.

Stage 4: Controlling

Despite meticulous planning and careful execution, projects fail if they lack control processes. In order to be able to identify any pitfalls and plan changes, you need to monitor and measure all the relevant activities.

Controlling will enable you to take any preventive measures and make any necessary adjustments to the plan, and, ultimately, keep your project on track.

To keep everything under control, you need to:

- Collect data from timesheets and completed tasks
- Compare the data against the plan (like task completion, budget, and time estimates)
- Check if you have achieved all the ambitious objectives

The ultimate goal of this phase is to meet the requirements and make your clients happy.

Traditionally, controlling process should includes the following activities:

- Time management
- Cost management
- Quality management

- Change management
- Risk management
- Issue management
- Procurement management
- Acceptance management
- Communication management

Stage 5: Closing

Project closure allows the team to evaluate and document the project. At this phase, it's always a good idea to hold a meeting, evaluate success, and discuss what you could do better in the future.

In the closing phase, you should:

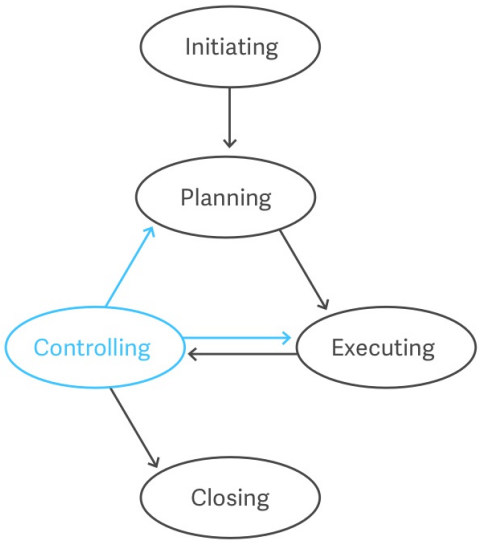
- Assess project performance (in terms of objectives, scope, deliverables, scope, schedule, and expenses identified during the previous stages)
- Rate how well your team carried out each project phase
- List project achievements and failures
- Share lessons learned for the future projects

The closing process is an excellent opportunity to praise any team members who contributed to the successful outcome.

Going back to previous phases

All the stages within the life cycle interact and are linked by the results they produce. Consequently, a project manager sometimes has to return to some earlier phases, make necessary adjustments, and then continue with other project management processes.

For example, during the controlling phase, if you need more resources for some task, you will have to go back to the planning stage and adjust the amount of resources planned for that task. Otherwise, this inconsistency can have a major effect on the entire project life cycle.



How stages flow on a project

Project manager roles and responsibilities

The strategic role of a project manager involves planning and organizing the resources and time necessary for successful completion of projects.

> *“A project manager is like a doctor who leads the trauma team and decides the course of action for a patient - both at the same time. Without the right kind of authority to efficiently handle all the project management issues, development teams can easily get into trouble.” - Scott Berkun, the author of “Making Things Happen”*

How did it all start? In the late 1980’s, Microsoft encountered a problem: the team didn’t know how to coordinate engineering efforts, marketing, and business department. They came up with a solution to involve an individual who will be given significant authority to be a leader and coordinator of the project that was later to become Excel.

Once Microsoft appointed a dedicated project manager, processes ran more smoothly and teams were more satisfied with work dynamics. Eventually, Microsoft adopted this new role and they named it “project manager”.

8 key roles and job responsibilities of a project manager

A good project manager has an excellent entrepreneurial mindset

which allows him to think about the project at large and beyond the basics of project management. Ultimately, they shepherd the overall effort and are responsible for the success and failure of the project.

A project manager has a critical role in making knowledge and information flow through the team seamlessly. They should be technical enough and have first-hand knowledge about the tasks he is assigning to others.

Technical understanding doesn't only enable them to communicate ideas effectively at all levels, but it also helps them gain team's respect. Since a project manager influences more decisions than anyone else in the company, earning employees' respect is the first thing they should do on.

There are eight key project manager roles and responsibilities:

1. Activity and resource planning

Many projects fail because of poor time estimates and weak assumptions. Planning is instrumental for completing the project on-time. The first thing a project manager needs to do is define the project's scope and determine available resources.

They should create a clear and concise document which he will use to guide both project execution and project control. Naturally, projects are not entirely predictable by nature and may need to be updated several times before they reach their final stages.

2. Organizing and motivating a project team

Instead of focusing their efforts on elaborate spreadsheets, long checklists and whiteboards project managers should be focusing on their teams. He is in charge of developing a plan that will support the team in reaching the goal and not hinder their performance. It is his job to steer the team clear of bureaucracy and stimulate them to show their full potential.

3. Controlling time management

Clients usually judge whether the project has succeeded or failed depending on whether it has been delivered on time - so on-time completion of the project is the prerequisite for success. A project manager needs to be able to negotiate achievable deadlines and communicate them consistently to his team.

He needs to develop a project schedule which consists of:

- Activity definition
- Activity sequencing
- Activity duration estimating
- Schedule development
- Schedule control

4. Cost estimating and developing the budget

A project manager should ensure that the project is completed within the approved budget. Even if a project meets client’s

expectations and is delivered within the predicted time scope, it may go wildly over-budget and still be considered a failure. A good project manager will frequently review the budget plan and forecast to avoid massive budget overruns.

5. Ensuring customer satisfaction

Customer satisfaction is undoubtedly the most important criteria when it comes to measuring project's success. One of the key responsibilities of every project manager is to minimize uncertainty, avoid any unwanted surprises and involve their clients in the project up to a reasonable extent. The easiest way to achieve this is to maintain effective communication and keep the clients regularly informed.

6. Analyzing and managing project risk

The bigger the project is, the more likely the project manager is to encounter many hurdles and aspects that don't fit into the initial plan. In this scenario, a project manager should meticulously identify and evaluate the potential risks. Also, he should develop appropriate strategies how to either avoid the risks or minimize their impact on the entire project

7. Monitoring progress

In the initial stage of the project, a project manager and his team have a clear vision and high hopes that they will be able to produce the desired results. However, the road to success is usually bumpy and fraught with challenges. When things don't go

according to a plan, a project manager needs to control and analyze both the team performance and expenditures and take necessary corrective measures.

8. Managing reports and necessary documentation

Finally, an experienced project manager will provide appropriate documentation where he will present final reports and identify areas for future development. It's the essential part of any project development and it has two core functions:

- To lay out the history of what has been done in the project, who has been involved in it and the time of its development
- To ensure that the project satisfies all the project requirements

Do you need a project manager?

No matter how big and demanding projects are, you need someone who will be in charge of efficiency and productivity on them. Research shows that 89% of high performing organizations have a project manager, and the profession ranks year after year as one of the most demanded profession in the 21st century.

Project management is indispensable to successful business and business owners should focus on finding a leader who will have the vision, the right skills, and knowledge to face the biggest challenges and ensure on-time completion of the project.

Project managers are integral parts of almost every kind of organization - from small agencies that have one project manager looking after a few projects to multinational IT companies which hire highly defined full-time project managers who are in charge of ambitious projects.

The many roles of a project manager

Many titles include project management in their job description. Does this mean that, for example, a product manager does the same job as a project manager?

Well, yes and no: product managers do work with the same material as project managers (like time and resources), but they have different goals and work under different constraints.

What's a project manager's job

A project manager is a term for someone who's responsible for making sure projects are completed on time and within budget. They are the ones responsible for execution and nothing is late.

"Project manager" is such a generic position that people often confuse it with other roles. To make things more confusing, project managers often end up with responsibilities outside of their traditional job description, like: working with clients, managing other project managers, or driving forward business strategy.

Job description also depends on company size: project managers in enterprises have clear and narrow job descriptions which is closer to the traditional job description; but in startups and agencies, their job descriptions vary wildly. Why? For a couple of reasons:

- Small companies can't afford to hire two specialists because there’s not enough work to justify two salaries, so project managers are expected to do everything
- Small companies lack processes and they learn by following what others do, reading blog posts, and customizing what they've learned so it suits their needs

Difference between managers

A project manager manages projects. That's all they do. They are in charge of meeting deadlines, staying within budget, and completing tasks. If they're doing things outside of those three things, they are more than project managers.

Whether you are a project coordinator or program manager depends on where you are on the organizational chart:

- **project coordinator**: you're an assistant to a project manager
- **program manager**: you manage other project managers

Once project managers get business responsibilities, they end up being product owners, product managers, business analysts, or account managers:

- **product owner**: your company works for clients, and you arrange user stories and represent the client in front of your team
- **product manager**: your company works on its own product,

you don't have clients (only customers), and you have to decide what features to implement

- **business analyst**: your company works for clients and you elicit requirements from them
- **account manager**: your company works for clients and your job is making sure clients are happy and bring profit

	Program Manager	Project Manager	Project Coordinator	Product Owner	Product Manager	Business Analyst	Account Manager
Manages project managers	x						
Is managed by a project manager			x				
In charge of execution		x	x	x			
Creates work for engineering					x	x	x
Techncial knowledge required				x			
Works with clients						x	x
Works with customers					x		
Makes decision based on data					x	x	
Deep customer understanding							x
Creates strategy	x				x		

Differences between management roles

All these positions appear similar because they require a similar skill set (organizing, people management and leadership, time management). If you have skills and experience as project manager, you can transfer those skills (although you'll still need some industry-related experience).

Project manager vs. project coordinator vs. program manager

The main difference between a project manager and a project coordinator and program manager is the hierarchy and the scale of responsibility: program manager manages > project managers,

who manage > project coordinators.

Project Coordinator

When a project manager has too many responsibilities, they delegate some of the work to a project coordinator.

Project managers spend 80% of their time on communication. Usually, a coordinator takes over some communication channels so the project manager doesn’t have to communicate with everyone.

For example, let's say a project manager has a local team of 7 people and a remote team of 20. Synchronizing the work of the remote team (which spans across different time zone and cultures) is a full-time job. By letting a coordinator manage the remote team, the project manager can make their work manageable and work with only 8 people (the local team and the coordinator) instead of 27 (everyone).

A project coordinator can take other responsibilities besides communication, like expediting tasks and other operational details. The project coordinator then becomes more of a traffic controller who arranges assignments and compiles status reports. This frees project manager to pursue other things, like identifying goals and aligning projects with business strategy.

Program Manager

A program is nothing else but a big project that's broken up into

several projects so it's easier to manager. At the head of a program is the program manager and their job is to manage project managers.

Program managers have a high-level overview, so it’s their job to communicate the vision and harmonize work of each project manager in the program.

Programs usually focus on some business objective or initiative. For example, you might create a program for rolling out a new product in your corporation; this is a big undertaking so you break it into several projects (eg. promotion strategy, distribution, logistics) and coordinate work.

Product manager vs. product owner

Product managers often work in product companies and have a more strategic role, while product owners often work in Scrum consultancies and they represent the client in front of the development team.

To develop a product, you need to satisfy the needs of your stakeholders: clients (if you're an agency) or customers (if you're a product company). To do that, you need to work with both your clients/customers (external) and engineering (internal).

Product owners are internally-focused and product managers are externally-focused.

Product managers lean toward working with clients/customers and product owners lean toward working with the engineering. In that sense, product owners are closer to being project managers than product managers.

Product manager's main concern is if the product they're developing will succeed in the market. Product owners, on the other hand, worry if the engineering has all the details they need to develop the product.

Product managers often work in product companies and product owners work in Scrum shops. So sometimes, the title says more about the company than about the actual job description.

It's not rare to see one person doing both jobs, which became a norm in startups and boutique dev agencies. But that creates a problem of balancing external and internal focus.

For example, if you spend too much time working with clients/customers, engineering will suffer: stories will lack detail, the team will have to wait for answers, and they'll produce less. On the other hand, if you spend too much time with engineering, the product will lose its touch with the market: you will have fewer client meetings, fewer sales materials, and you'll end up developing a wrong product.

Product Owner

The product owner is a role in Scrum (which is project management methodology). Product owner decides how valuable

each backlog item is and what the team will work on next.

Product owners are there to represent the client's interest in front of the team. When someone on the team has a question, they ask the product owner who must think and answer like the client would if they were personally there.

Product owner's other responsibilities are to: describe each user story in detail, participate in daily standups, and make sure the product backlog is visible, transparent, and clear to everyone.

Product Manager

In contrast to the product owner, product managers have a more strategic role. They know the market, what their customers need, product position, and based on that they create the product roadmap.

Also, while a product owner represents the client, a product manager represents the customer. Because a product manager has to listen to thousands of stakeholders (and not just one client), they need to do a lot of data analysis and draw on insights from other departments like support, marketing, and sales.

Business analyst vs. account manager

The role of a business analyst and an account manager is almost the same: they both work with clients and their only contact with engineering is when they hand off requirements.

But there's a subtle difference: business analysts focus on the technical aspect and account managers on the people or financial aspect (whether account manager schmoozes clients or spreadsheets depends on the industry and its particular definition of account manager's role).

Business Analyst

Business analyst is the link between clients and engineering. Their job is to understand both business and technology, and then prepare user stories, high-level functional diagrams, etc.

For example, a client might need new software to they can be more productive but doesn't know how to develop it. So they hire a consultancy, which then sends a business analyst who talks to the clients, sees how they work, and define requirements for them.

Business analyst's' job is to elicit requirements from the stakeholder, analyze and document those requirements, and hand them off to the product owner.

Account Manager

Account manager handles accounts and does whatever it takes to make sure the client is happy. They have a deep understanding of the customer's business and know political and environmental factors that affect it.

While business analysts are more prevalent in IT industry, account

managers can be found anywhere, as virtually every business depends on accounts.

An account manager evaluates progress against dates and budgets like a project manager, but they put a greater emphasis on financial indicators (like sales, profit, and overhead); which explains why most account managers usually have a Profits and Losses background.

Because account managers are responsible for company profits, they have more power when it comes to planning prices and incentives (as opposed to project managers who are constrained by technical requirements and have to correct deviations by changing priorities and juggling resources).

Do you need a project management certification?

You've probably heard about all sorts of project management certifications and wonder whether it's worth having one. It depends:

- If you want to advance your career, you should get certified.
- If some organization uses some project management framework, you need to get certified.
- If you want to manage projects better, certification won't help you much.

The most popular project management certifications

A certification is different from a degree. Degrees are awarded by universities and governments, take much longer to complete, and require students to take general education courses in addition to profession-specific ones.

Certifications, on the other hand, are awarded by professional organizations and focus only on one area. A certificate doesn't necessarily mean a person has sufficient knowledge, only that they passed the test. Note that having a certificate doesn’t mean you’re legally able to practice or work in a profession - that's licensure.

When it comes to project management, PMP is the most popular

certification (and most widely recognized). After it, there are PRINCE2, SAFe, CAPM, ITIL, CSM, and others.

Each certificate is different. Some are only for the IT industry, while others are applicable across industries; for some job, you need a certificate (because the company uses the framework), while for some jobs a certificate is just a nice-to-have.

PMP (Project Management Professional) certificate covers project management in general. Because it doesn't tie you down to a specific industry, it's the most popular choice for project managers. There's also CAPM (Certified Associate in Project Management), which is a lite version of PMP and is easier to obtain (they are both issued by PMI).

PRINCE2 (Projects IN Controlled Environments) is a project management methodology for which you can get a certificate, which issued by the government of UK. While PMP is about project management in general, PRINCE2 is highly prescriptive and has strict procedures every project must follow. If you get the PRINCE2 certificate, you will have to either get a job at a company that already manages projects using PRINCE2 or introduce it. If you want to keep your options open, PMP is the better choice.

ITIL (Information Technology Infrastructure Library) covers a larger area than PMP. ITIL is about the whole service lifecycle, and project management is just a part of it (the course also covers Service Strategy, Service Design, Service Transition, and Service Operation). ITIL is applicable only in IT services industry.

CSM/PSM (Certified Scrum Master/Professional Scrum Master) are two different certificates (issued by different organizations) that both cover Scrum, the most popular agile framework in software development. If you know you want to be a Scrum project manager exclusively, getting a Scrum certification is a safe choice.

SAFe (Scaled Agile Framework®) takes Scrum and applies it to enterprise (given that Scrum works well for teams but doesn't scale when it comes to the company as a whole). SAFe is used in large corporations who have software-intensive projects, where teams are highly interdependent.

Pros of having a project management certification

Easier to get a job

Having a professional certificate looks good on a resume. It means you are vetted and have the knowledge required for the position. When competing with hundreds of others resumes, it's easier to stand out.

Plus, it's good for specialization. If you specialize in Scrum, for example, you'll narrow your options but increase the likelihood of getting a job. Having a particular certificate means you've chosen your field and are dedicated to project management as a profession.

> *I don't believe I ever missed out on a project opportunity because I did*

not have the PMP Certificate. However, now that I'm no longer with the same company and I'm marketing my skills to prospective employers, I know I have been passed over due to the lack of those initials on my resume. One Sr PM on a government project interviewed me and recommended me to the decision makers, but when he got back to me, he said they decided to go with someone who had a PMP certification. - David Pickles

Better pay

People with the PMP certificate make $6,000-10,000 more per year than people without PMP. A certificate is your bargaining chip for negotiating higher compensation, which can earn you 16% bigger salary.

Community

Certificates promote a certain philosophy and build communities. This advances project management as a science and gives you a common language so you exchange experience with other professionals and improve your skills.

For example, to renew the PMP certificate, you need to collect points by visiting workshops, volunteering, and being part of the community (which forces you to meet other project managers and exchange knowledge).

Cons of having project management certification

Takes time and money

Each certification program is structured differently. For PMP, you need to file your entire project management experience to pass the vetting process. Then, you have to spend 35 hours listening to the course and pay $405 for the certificate ($555 if you're a non-PMI member). Exam prep courses will cost you somewhere between $1,500 and $2,000, plus roughly $1,000/year to acquire points necessary for certification renewal. All in all, a 40-year career cost of the PMP certification will cost you $42,500.

Other certificates have different cost structures. Some have a high cost of training, while others cost more to acquire the certificate itself and renew it each year.

Little practical application

98% of the day-to-day project management activities don't require anything they teach in a course, as PMBOK and other frameworks work only on megaprojects. But when you run hundreds of small projects, all you need are organization, people skills, and common sense.

Also, some frameworks are too prescriptive and don't allow you to customize it your particular workflow. And some courses are dated, as they focus exclusively on waterfall methodologies (like PMI's).

> *As far as I'm concerned, a PMP certification proves you're "book smart." The PMP does require some real world experience, but it's only three years—just barely a start in a career. The real skills that make or break*

> *a project manager is the emotional intelligence to know which tool to use at the right time, including a deep respect and appreciation for human behavior and group dynamics. Human behavior is not a work breakdown structure, and methodology alone will not get you there. - Pam Stanton*

Not a true reflection of skill

A certificate is just a sign you can pass the test. You can be the worst project manager without a single successful project behind you - yet it doesn't stop you from having a certificate. Because of this, some companies disregard certificates and rely on track record and personal references.

Plus, there are a lot of organizations that provide training to extract as much money as possible.

Certificates lead to bureaucracy

The biggest problem with project management certificates is that they can impede progress. They introduce superfluous concepts and processes which make everything complicated

> *It's like writing a 30-page document on how to make a peanut butter sandwich. I would be very much in favor of a rewrite of the book (PMBOK), and consolidate and streamline the information into a more coherent framework. - Dave Schroeder*

For instance, to make sure a project adheres to PMBOK/Scrum, a project manager has to create documentation for the sake of

documentation. Teams then start looking at requirements as a paper drills so they can check the project off as “complying.”

What happens in most cases, instead of creating a new risk management plan, the team finds a previous project plan, change the name, and put it in the archive. Same with other documents. The problem is, most of the documentation will never be used, it costs a lot to make, and delay the project.

Project managers end up being paper pushers and stop progress - which is opposite of what a project manager needs to be doing. Some frameworks start with agile principles, but ironically end up anti-agile.

Certificates - yes or no?

Getting a project management certificate depends on your career aspirations and the industry you work in. If you work in construction, finance, government, or healthcare - industries with a lot of bureaucracy - having a certificate is a plus.

In small companies, a certificate is a hindrance as your knowledge won't be suited for the type of projects they have. You're much better off using a simple Kanban workflow than having a complex methodology no one understands.

Project management certificate is not a silver bullet that will make you an excellent project manager. It's merely a sign that you have the basic knowledge of project management processes, as

described in a course book.

Introduction to project management methodologies and frameworks

The roots of modern project management

People began formally managing projects in the 1950s. Before that, projects were managed on ad-hoc basis using Gantt charts and other tools. But in 1950s, corporations and the US government decided to manage project more systematically; so, they invented two project-scheduling models - CPM and PERT.

Both models were used for the same purpose (estimating and planning activity), but for different types of projects: if you had a project with a lot of moving parts and high unpredictability, you'd use PERT; otherwise, you'd use CPM.

The introduction of those two models is very important because it marks the shift of mindset from "every project can be managed using the same tools" to "each project is unique and requires a certain set of tools to be successful".

Once we began studying project management as a science and discovering project archetypes, we were able to come up with recipes: if you are run a type A project, you should use these tools; if you run a type B project, use these tools instead.

Back then, there were no formal methodologies like today. Instead, you had a few approaches and you'd use one over the

other, depending on what kind of a project you had.

As project management spread across a wide variety of industries, we had to tailor each approach to our specific industry and its type of projects. For example, agile methodologies are great in software development but they don't work in construction or health care.

After some time, we began bundling methods and techniques into methodologies. You can think of methodologies as surefire recipes we follow so we can run our projects consistently on time and within budget - consistently being the key word.

Why there are so many project management methodologies

Just because some methodology worked on one project doesn't mean it will work on another. So we customize a methodology, slap a new name on it, and begin selling it as a new silver bullet to all our problems - only to end up with the same problem: the methodology works on some projects, but not on others.

In addition to methodologies, we also create frameworks. They are more prescriptive than approaches (like waterfall or agile) yet more flexible than methodologies.

Each methodology and framework is suited for a different type of project. Because there are many types of projects, there are many types of methodologies and frameworks. It’s in our human nature

to want certainty and magic formulas (in a field where that isn't possible), so we continue creating new methodologies and frameworks each day.

Each methodology and framework comes with its unique set of strengths and weaknesses:

- some are great for enterprises, while others for small autonomous teams
- some are optimized for projects with a lot of variables and uncertainty
- some are for projects where requirements change frequently
- some are for developing a product while others for providing a service
- some focus on controlling costs and some focus on controlling time
- some are for software developers while others for other departments (eg. marketing)

The difference between a methodology and a framework

In project management community, there is no agreed understanding whether something is a methodology or a framework. There is an agreement on definitions (what is a methodology and what is a framework), but when it comes to applying those definitions, there's a big disagreement. For

example, some experts label "Event Chain Methodology" as a framework even though the name says it's a methodology.

To further complicate the whole thing, professional project managers use the words methodology and framework interchangeably.

Here is the main difference between a methodology and a framework (at least in theory):

	Framework	Methodology
About	What to do	What, when, & how to do
Consistent outcome	Low	High
Customizable	Yes	Somewhat
Expertise needed	High	Medium
Implementation difficulty	Medium to High	Medium
Provides metrics	Maybe	Yes
Underlying philosophy	Maybe	Yes
Structure	Flexible	Prescriptive

Framework vs methodology

A framework provides structure and direction on a preferred way to do something, without being too detailed or rigid. They provide guidance while being flexible enough to adapt to changing conditions or to be customized for your company while utilizing vetted approaches.

A methodology is an approach to doing something with a defined

set of rules, methods, tests activities, deliverables, and processes which typically serves to solve a specific problem. Methodologies demonstrate a well thought out, defined, repeatable approach.

As you can see, the difference between a methodology and a framework is in the level of granularity: the more rules it has, the closer it is to being a methodology on the framework-methodology spectrum.

There’s a running joke in the project management community that illustrates the nature of methodologies: "The difference between methodologists and terrorists is that you can negotiate with terrorists".

In contrast to a methodology, a framework doesn't give you answers; rather, it guides you through a set of questions so you can develop your own solution and policy.

This all ultimately means there is no official classification which you can consult to get a comprehensive overview of all the project management methodologies and frameworks out there. And creating such a thing is impossible. Each classification you’ll run across is arbitrary (just like the one you'll find here) and it's your job to piece everything together and make sense of it.

How to use project management methodologies and frameworks

In the end, whether something is a methodology or a framework

doesn't really matter. What matters is how far you take the concept and whether it works for your particular project.

If you take a method and create a system around it, you've created your own methodology. In that case, congratulations! Now you can promote it, teach it, issue certifications for it, and make a nice living as a project management consultant.

Methodologies/frameworks are only good if you're practical about them and implement stuff within reason. If you take them off the shelf and force them upon the business, it will end badly. This happens all the time with ISO 9000, ITIL, and similar certifications.

On the other hand, if you actually learn and understand the methodology/framework, adapt it to fit your needs, and roll it out properly, you'll end up with something valuable.

The main problem with methodologies happen when companies jump on board because something is a hot topic at the moment. You need to understand that implementing a methodology/framework takes years of hard work.

For example, the reason a lot of people think Scrum sucks is because a lot of companies jump into it head first, buy a ton of "Scrum products" and force everybody to change their terminology. Then things get broken, while the company keeps bragging how they are a Scrum shop.

Implementing a methodology/framework does not mean throwing

all your existing processes out the window. Instead, it's about slowly changing company culture and the way you think and work.

Note: when you hear someone uses waterfall or agile methodology, it doesn't mean they use a methodology called waterfall - instead, they use word “waterfall” as a shorthand or an adjective, meaning they plan the whole project upfront and then execute.

How to choose the right project management methodology

There are so many project management methodologies that it can be difficult to choose the right one.

But here's the secret: you don't really need a project management methodology.

Project management methodology is nothing but a definition of the way you work. If you already have some workflow in place (even if it's not optimal), you don't need to burden yourself with complicated methodologies.

The truth is, 90% of all project management methodologies out there are used for managing enormous projects, justifying consulting fees, or both. The chance is, if you're running a small business or an agency, you'll only waste time and energy by learning what is PRINCE2 or how to create a PERT chart.

Although you don't need a methodology, you could still use learning some project management concepts. Learning what is buffer time or why you should limit work-in-progress can greatly boost your productivity.

Our advice?

- Read about Theory of Constrains, Critical Chain, and a few other books

- Learn how to use a Gantt chart
- Adopt a light-weight project management process (like Kanban or Scrum)
- Invest in a good project management tool

Everything else is unnecessary.

A brief overview of general project management methodologies

Traditional Project Management

Traditional project management is the most common way people manage projects. It's not a methodology but a collection of techniques, like WBS (work breakdown structure), dividing a project into phases, Gantt-charts, etc.

PERT Network Chart

PERT network chart is an advanced tool used for calculating how much time each activity can take so you can see what tasks are allowed to be late. It's a useful tool, but in practice it takes too much time, plus someone on the team needs to update it constantly for it to be useful. PERT is mostly used on huge, complex projects with activities where time is unpredictable.

Critical Path Method (CPM)

Critical Path Method is very similar to PERT, only CPM doesn't take

into account multiple times (optimistic, pessimistic, optimal). As such, CPM is used on predictable projects, primarily to control cost (and not time, as in PERT).

Critical Chain Project Management (CCPM)

In Critical Chain Project Management, you calculate which tasks are critical for project completion (like in CPM and PERT), while taking into the account existing resources. This is a useful methodology to know as it can help you better plan and estimate projects, but it's not very practical as there are easier methods that are more convenient (and easier to learn) in everyday work.

Adaptive Project Management

Adaptive Project Management lets you change project plan as you go along and get feedback. It's used on huge projects where requirements are uncertain. It's similar to Agile (which is used on software development projects).

Extreme Project Management (XPM)

Extreme Project Management (XPM) is almost exactly like Adaptive Project Management, only created by a different team of consultants.

Six Sigma Methodology

Six Sigma Methodology is used in manufacturing to keep number of defects low. As such, it's not really useful in project management (the only reason we include it here is because you

can often hear about Six Sigma in context of project management, and that is because in manufacturing a project means something else).

Theory of Constraints

Theory of Constraints is a methodology which helps you identify bottlenecks and solve them. The whole idea is that bottlenecks limit how much you can accomplish and you should focus all your energy on fixing them. Like CCPM, it's a useful concept to know about, but it won't help you manage projects (the workflow is more suited to manufacturing).

PRINCE2

PRINCE2 is the only full-fledged project management methodology so far. It describes how all your processes should look like, in great detail. It clarifies deliveries, ensures processes are repeatable, defines roles and responsibilities, etc. As such, it’s great if you run large, predictable, enterprise projects. On the other hand, for everyone else (including agencies), PRINCE2 is too complicated to be useful.

A brief overview of project management methodologies used in software development

Waterfall vs Agile

There are two basic ways to manage software development projects: waterfall and agile. Waterfall works the same as

traditional project management (plan everything upfront using Gantt charts). In Agile, on the other hand, you don't make big plans - instead, you iterate until you get it right.

There's nothing special about waterfall. It's the same as traditional project management, only used in context of software development, and to serve as contrast to Agile.

Agile is best suited for small-to-medium size organizations because the less people there are, the easier it is to make a decision and respond to change. Also, agile is more suited for product companies than agencies.

Agile isn’t the right approach for every software project. If you don’t have access to customers, can’t iterate, or if you have complex organizational structure, it’s difficult to adhere to agile principles.

Note that Agile is only a set of principles and there are no processes you can follow to manage projects. Instead, you'll have to use one of the agile methodologies or come up with your own workflow.

Scrum

When people say they use agile to manage projects, they usually mean they use Scrum. Although Scrum is mostly used on software development projects, it can also be used in education, government, and other industries. Scrum is a framework, which means you can adopt it to suit to your workflow. If you run

complex or semi-complex projects, Scrum is a good choice.

Kanban

Kanban is the simplest and most flexible methodology. It works for everyone, from teams in enterprises to small agencies (it even works as a personal productivity tool). All you need to get started is a Kanban board so you can see at what stage each task is, and that's it. If you need more, Kanban also has advanced concepts such as controlling bottlenecks and more. Kanban is the most versatile methodology, so if you're new to project management, start with Kanban.

Extreme Programming (XP)

Extreme Programming (XP) is very similar to Scrum, only it's designed specifically for pure software development. Unlike Scrum, XP prescribes development processes like test-driven development, refactoring, pair programming, continuous integration, coding standards, etc. If you run purely software development projects, XP is a good alternative to Scrum. It works similar to Scrum (so your team won't have trouble adjusting), contains a lot of best development practices, the community is big, and there are a lot of good resources to get started.

Rational Unified Process (RUP)

Rational Unified Process is a complicated agile methodology. It's a cross between Scrum and waterfall (as you divide a project into phases and plan for iterations). But because it's formalistic, heavy-

weight, and highly-ceremonious, it's mostly used for large, complex, and long-term projects - ie. projects in enterprises.

Crystal Methods

Crystal Methods is not a single one-size-fits-all methodology but a family of several methodologies where you use one of the methods depending on the size of your team. The downside to Crystal Methods is that learning it is not easy: there is no quick and easy guide, the community is very small (compared to Scrum), and it's difficult to find an expert.

Feature Driven Development

Feature Driven Development is similar to Scrum, and serves as a great stepping stone for companies who want to transition to Agile. FDD invests more time in upfront requirements and has well defined steps (like in Waterfall), while still following core agile principles.

Joint Application Development

Joint Application Development is not a full methodology but more a technique. It's used for defining software requirements through a series of workshops that last for a few days, called JAD sessions. It's used in big IT companies who use more traditional methodologies (the whole point of a JAD session is one big specification).

Why agile doesn't work for digital agencies

Although agile is a huge buzzword, the truth is, agile doesn't really work in an agency setting. Agencies regularly deal with new technologies and vague requirements, and struggle with accurate estimates so you'd think agile would be perfect. But it isn't because clients prefer the predictability of the deliverables, budgets, and timeline that only waterfall can give.

The client wants to know how much it's going to cost and when it will be finished. All agile can say is "we'll see", which is not something a client wants to hear.

The flexibility agile gives is very expensive. If you pivot and change plan, as agile encourages you to, you still need to bill your client for spent time while giving them little in return.

Another problem with agile in an agency setting is that to be truly agile, you need constant access to the client and they need to provide feedback on the fly to keep the project moving. This is not possible. The most common reason why projects are late is because agency has to wait for client feedback. Agile would only make deadlines worse, while the upside - what the client really needs - is not that important.

In the end, whether agile can work for you depends on your clients and the trust between: do clients trust you to deliver and are they willing to pay for failures on the path to success? The most common answer is no.

Agencies want to get paid for their time, and clients want

agencies to deliver best work right at the start. And that is the main reason why agile doesn't work for agencies.

What’s the best methodology for digital agencies?

Out of all the project management methodologies, only a handful of them are applicable for digital agencies.

If you run a marketing/design agency

If you run a marketing/design agency, all you need is Kanban and a few techniques from traditional project management. Scrum can also work (like when you're working on a long-term project) but that's very rare because most agency contracts are finished after 3 iterations, and the client won't like the idea of getting half-finished work and wait while you iterate.

If you run a development agency or a consultancy

If you run a development agency or a consultancy, you can use Kanban, Scrum, or XP, with this caveat: if you work on short-term, on-and-off projects, Scrum and XP will be a tough sell for the client (although you can still use some best practices from each methodology).

In a nutshell you won't go wrong with Kanban + a few traditional project management techniques, no matter what types of projects you have. Further, if you work on big, complex, and long-term software projects, you can adopt to Scrum or XP to boost productivity even more.

Does that mean there's nothing useful to learn that can help you better manage projects?

Not true. Although you shouldn't waste time with project management methodologies, there is still a lot you can learn about project management that will help you increase productivity (like planning poker, buffer time, critical chain, limiting bottlenecks, RACI matrix, etc.).

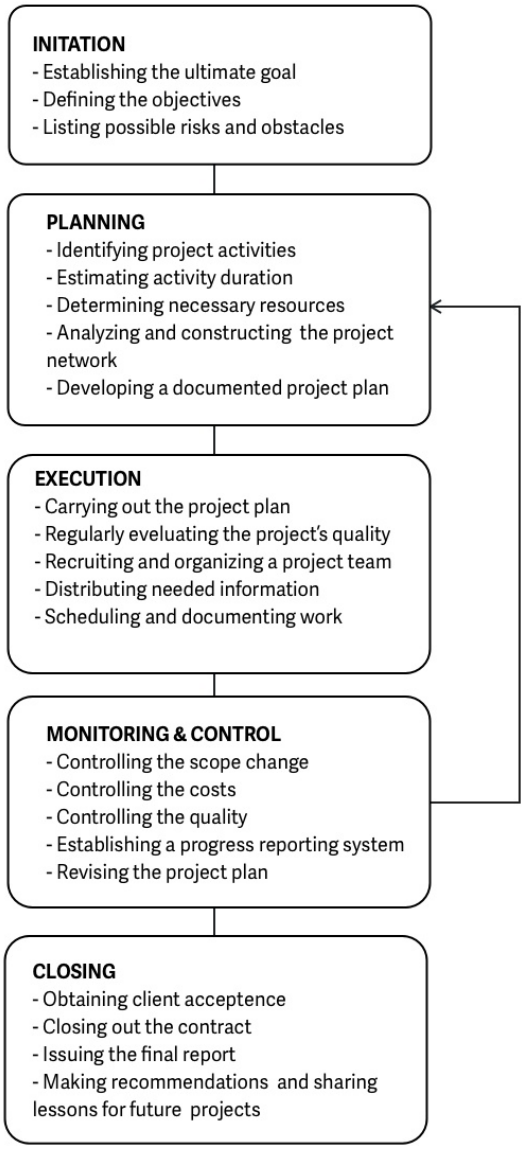
The point is, don't overburden yourself with methodologies but instead learn general techniques and tools that can be used regardless of the methodology, and see if they are applicable to your work.

PART II

General Project Management Methodologies and Frameworks

Traditional Project Management

Traditional project management is a universal practice which includes a set of developed techniques used for planning, estimating, and controlling activities. The aim of those techniques is to reach a desired result on time, within budget, and in accordance with specifications.



Project life cycle stages

Traditional project management is mainly used on projects where activities are completed in a sequence and there are rarely any changes.

The concept of traditional project management is based on predictable experience and predictable tools. Each project follow the same lifecycle, which includes five stages: initiating, planning, executing, controlling, and closing.

The beginnings of traditional project management

Although project management was first introduced as a discipline during the 1950's, it has been around for thousand of years and has been used in creating some of the biggest projects, from the Great Pyramids to the Transcontinental Railroad.

Those large-scale projects changed the face of the history and mankind forever. However, as time went by, business owners and entrepreneurs found it hard to keep up with the fast pace of technological development and ever-increasing demands of the market.

Business leaders realized that they needed a system that will help them manage large-scale projects. They needed a well-structured methodology that would help them bridge gaps and ensure a consistent work pattern.

As a result, traditional project management was developed. Its ultimate goal was to make sure all the tasks are carried out in

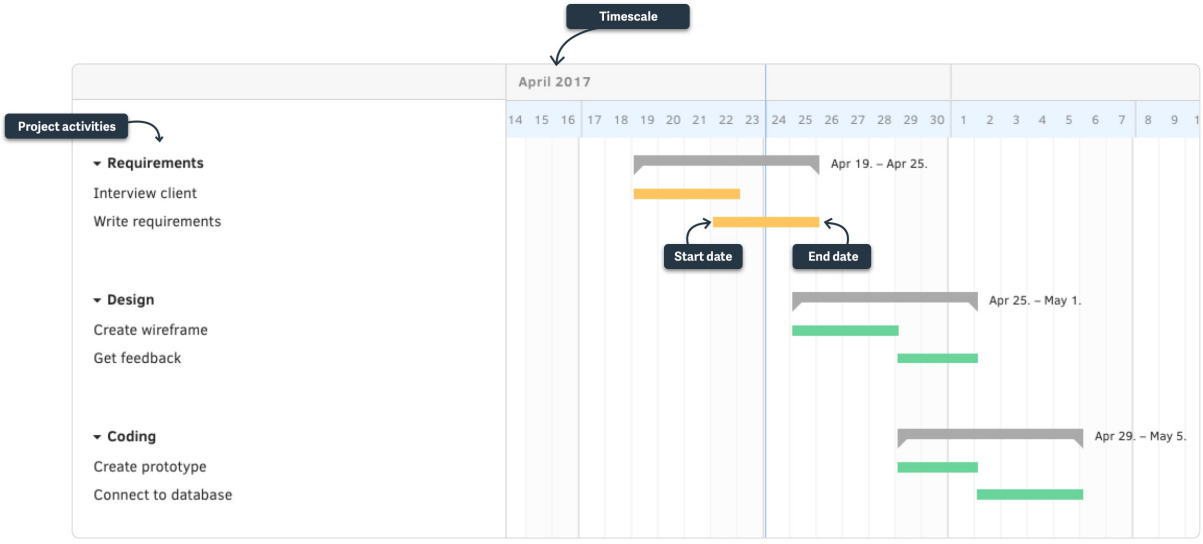
predetermined orderly sequence.

Gantt chart

Gantt chart is the most important technique in the traditional project management. Its creator was Henry Gantt, which is why he's considered the father of traditional project management.

Gantt chart gives you a simple overview of a project. It is one of the most useful ways of presenting tasks and activities of the project on a timeline.

In the beginning, people used to create Gantt charts by hand or by using magnetic blocks and Lego. As technology progressed, we began making Gantt charts in Excel in a matter of seconds.



Most common way of planning waterfall projects using Gantt chart

Left part of the Gantt chart shows the project activities and the top shows the time scale. Each activity is presented with a bar. A bar's

position tells us about the beginning, duration, and end of the activity.

By looking at a Gantt chart, we can learn:

- What the project tasks are
- Who is working on each task
- How long each task will take
- How tasks overlap and link with each other
- The start and finish date of the project

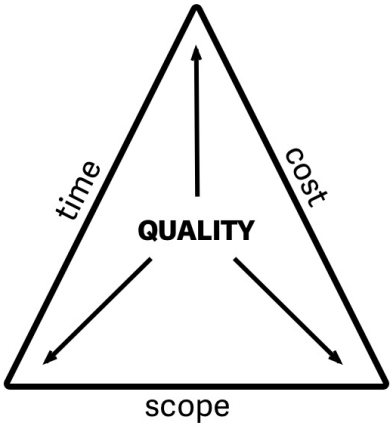
Today, we mainly use Gantt Chart to track project schedules and make project management less stressful. Gantt chart helps us understand the relationship between tasks more clearly, keep all the team members on the same page, and successfully complete project.

Gantt Chart comes with a lot of benefits: it lets you organize your thoughts, gives you a clear layout of what you're doing, helps you set a realistic time frame, and it's highly visible.

However, if you are working on an ambitious and demanding project with hundreds of tasks, charts might become too complex and make you feel overwhelmed. For example, as things change, you need to update the Gantt chart. Also, the size of the bar does not necessarily indicate the amount of work needed to successfully complete the project as activity may require more resources than you initially expected.

What is project management triangle?

There are many project constraints and the three most frequent ones are time, cost, and scope. They are a part of every projects and together they make up the project management triangle.



Project management triangle

Scope

In the initiation phase, it's important to specify all the steps of the project development including what WILL be was done and what WILL NOT be done. If you want to keep the project under control, you need to allocate some time and carefully plan and define the scope.

Time

Time is an invaluable resource. While we can control the processes and make necessary changes to improve them, we cannot control time. One of the biggest challenges each project manager has to face is to use time efficiently, keep the project on schedule, and reach the desired objectives.

Cost

As a project manager, you should define the budget in the early stage of the project and then compare it with the figure your customer initially offered. If the client decides to spend a certain amount of money that doesn’t match your requirements, you can prepare a business proposal which will include the estimates of the total cost of the project. This proposal helps the customer base his decision on more accurate estimates.

All three constraint are interconnected and depend heavily on one another. Once you reduce the time allocated for the project, the cost increases. Also, the scope of the project dictates the pace and a number of resources necessary to realize and successfully complete the project.

While it’s virtually impossible to keep your eye on everything, it’s an imperative to maintain the balance and strive towards the equilibrium of the triangle no matter how challenging it can be.

The area inside the triangle represents the quality and it’s the ultimate objective of every project delivery. A good project manager will find the way how to control all three constraints of the triangle and produce the best quality.

Project manager's ultimate goal is to meet their customers’ requirements and live up to the quality standards. In order to do so, you as a project manager need to control both the quality and the constraints.

Building on top of traditional project management: quality management

In order to improve the quality of both product and project management, you should integrate the following tools:

- Continuous quality management
- Process quality management

Continuous quality management is used to analyze any gaps where certain processes and steps could be carried out to improve the quality. You can undertake any number of improvements and then check them against the targeted improvements. It is cyclical and mostly present in customer-driven organizations.

Process quality management involves factors that have a major effect on the success of business processes and is based on the assumption that the organization has documented its mission and vision.

Using a grading system, all the process that are critical to a successful business are identified, analyzed and related to continuous quality management. The results of the analysis help the project manager make necessary improvements for given processes and the next stage is the initiation of project management life cycle.

Traditional project management in the 21st century

The computer and the Internet have become essential parts of almost any business. The computer replaced most manual jobs and new professions such as IT engineers, programmers, and project managers started emerging.

The processes became more complex and demanding, and the traditional project management no longer offers the best solutions to business problems. The concept of traditional project management has been changed and extended through different project management methodologies and frameworks.

Nevertheless, traditional management is still considered the foundation of all modern approaches and is still the prevailing methodology when working on big construction projects.

Critical Path Method (CPM)

CPM (Critical Path Method) is a mathematical algorithm that helps you analyze, plan, and schedule complex projects.

Each project consists of a number of tasks and activities that are interconnected and essential for project's success. The more complex the project gets, the more demanding the project management is. That's where CPM comes in hand.

At its core, CPM is a powerful tool that allows you to identify the longest path of planned tasks necessary to meet the deadlines and identify the early start and finish dates.

By determining the critical path, you will know which activities are critical in completing the project, and which ones won't have any serious impact on the project development and can be delayed.

How was CPM created?

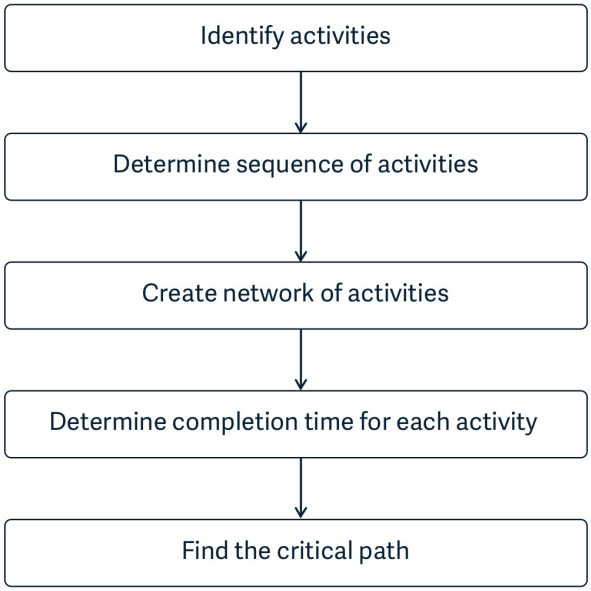
In the late 50's, El DuPont de Nemours Company, an American chemical company, was seriously falling behind its schedule, and they needed something that would get them back on track.

They came up with a solution to divide their project into thousands of tasks, measure the time each task will take, and how asses critical they are to the entire process. They called this technique Critical Path Method or CPM.

CPM was first tested in 1958 in a project to construct a new chemical plant, and has ever since, been one of the most frequently used techniques of project management.

Steps in critical path method

A critical path method includes the following steps:



CPM steps

1. Identifying activities

By using the project scope, you can break the work structure into a list of activities and identify them by name and coding; all activities must have duration and target date.

2. Determining sequence of activities

This is the most important step as it gives a clear view of the

connection between the activities and helps you establish dependencies as some activities will depend on the completion of others.

3. Creating a network of activities

Once you determined how activities depend on each other you can create the network diagram, or critical path analysis chart; it allows you to use arrows to connect the activities based on their dependence

4. Determining completion time for each activity

By estimating how much time each activity will take will help you determine the time needed to complete the entire project. (While with smaller projects you can make estimates in days, more complex ones require making estimates in weeks)

5. Finding the critical path

A network of activities will help you create the longest sequence of activities on the path or the critical path using these parameters:

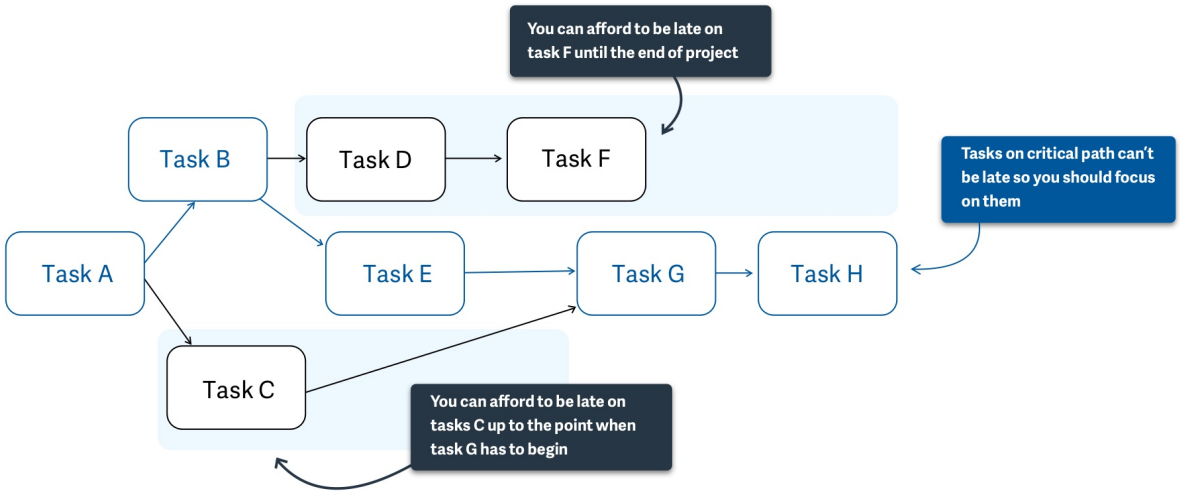
- **Early Start ES** - earliest time to start a certain activity providing that the preceding one is completed.
- **Early Finish EF** - earliest time necessary to finish activity
- **Late Finish LF** - latest time necessary to finish the project without delays
- **Late Start LS** - latest start date when the project can start

without project delays

If there is a delay in any task on the critical path, the whole project will have to be delayed. The critical path is the path where there can be no delays.

Naturally, not all the project activities are equally important. While some have a huge impact on the critical path and are therefore critical, others don't make much difference to the project if they are delayed.

The critical path method helps us determine which activities are "critical and which have "total float". However, if any of the floating activities get seriously delayed, they can become critical and delay the entire project.



How CPM works

Example: using CPM on a construction project

Take home building project for example. In order to build a house,

you need build the foundation first. Construction is a sequential process where all the activities from clearing the lot to building the roof follow one another and are dependent on each other.

If you don’t frame the house, you cannot insulate; if you don’t do drywall, you cannot paint.

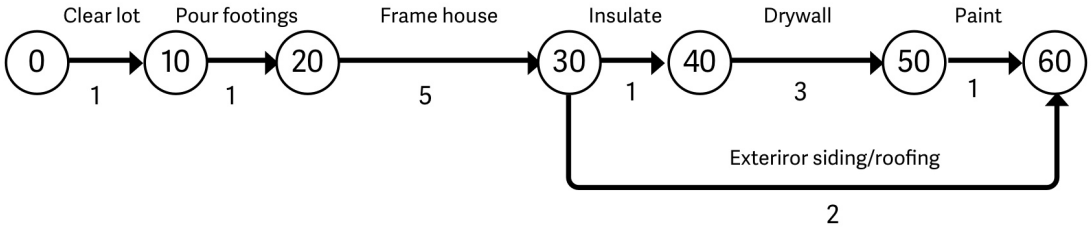
Critical path method allows you to first identify all the activities, then figure out how long each will take, and finally determine the longest path, also known as the critical path.

Once you determined the activities of building a house (clear lot, poor footings, frame house, insulate, drywall, paint, and roofing), you need to establish their dependencies. For instance, while you cannot frame the house until you have poured the footing, exterior siding and roofing can take place at the same time.

Although some activities can happen at the same time, all activities must finish at the designated time identified by the critical path.

The next step is to identify the time necessary to complete each activity. While painting may take 7 days roofing will take only 3 days.

The fact that floating time is not on the critical path makes all other activities critical. That is, if one activity on your critical path is 5 days late, it means that the entire project will be 5 days longer than expected.



An example of CPM diagram on a house construction project

How resource limitations affect the CPM

As much as we would like to take the best out of the critical path method and make our projects run smoothly, there are still certain limitations that affect our projects and create new dependencies. For example, if the number of team members suddenly drops from 10 to 7, you face resource limitations (ie. resource constraint).

In such scenario, the critical path changes into “resource critical path” where resources related to each activity become an integral part of the process.

This means that some of the tasks will have to be performed in a different order which may cause delays, and, consequently, make the project longer than expected.

How can you benefit from CPM

Although critical method path may have become an outdated technique due to fast-paced technological advances, it still offers a number of advantages:

- Prioritized tasks
- Clear insight into your project’s timeline so you can reduce

the time necessary for project completion

- You can compare planned vs actual progress
- Easy risk assessment
- You can redistribute your team members more efficiently
- Helps your team stay focused

The critical path method lets you stay focused on the big picture by giving you a clear view of all project activities and its potential outcome.

The best thing about CPM is that you can reschedule less important tasks and focus your efforts on optimizing your work so your team can avoid delays.

Program Evaluation and Review Technique (PERT)

The Program Evaluation Review Technique, commonly known as PERT, is a visual tool in project planning that helps organizations analyze and represent the activity, and evaluate and estimate the time required to complete the project within deadlines. PERT allows planners to identify start and end dates, and ultimately reduce costs and time needed to complete the project.

PERT was developed in 1958 by the US Navy as part of the Polaris project. Their aim was to manage the Polaris submarine missile program. At around the same time, an American chemical company El DuPont de Nemours devised a very similar approach called Critical Path Method (CPM).

Unlike CPM, which determines the longest path needed to complete a project, PERT gives you three different time estimates. While CPM focuses on time, PERT focuses on time-cost trade-off.

PERT in 5 steps

Step 1. Identifying specific activities and milestones

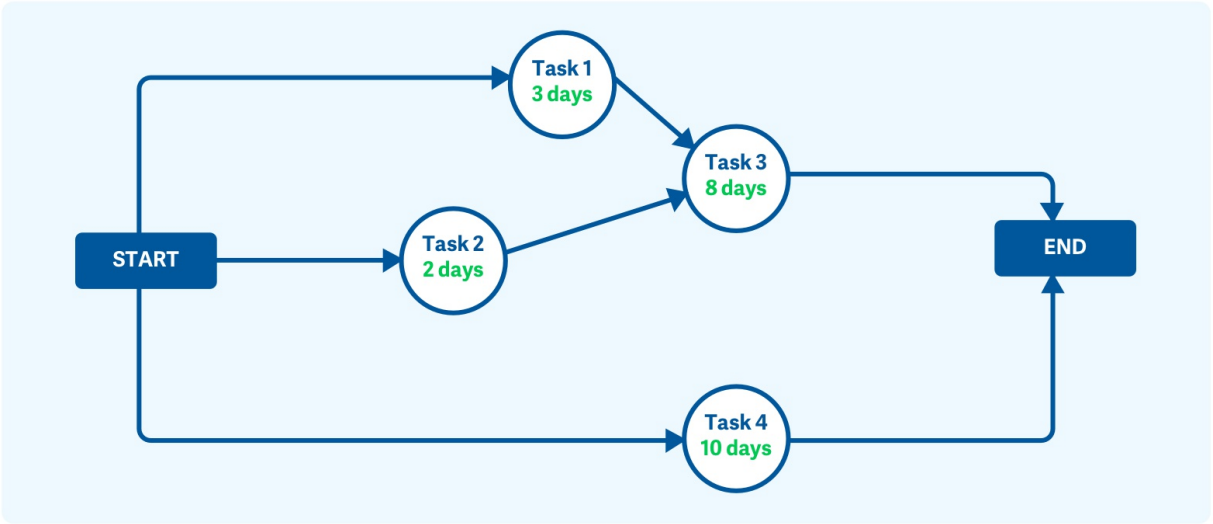
By listing all your tasks in the table, you get the clear overview of all the steps which you can subsequently expand by adding information on sequence and the time necessary to complete each activity.

Step 2. Determining the sequence of activities

While it easy to predict the order of some activities, other tasks may require more in-depth analysis which will help you determine their order more easily

Step 3. Constructing a network diagram

Once you established the sequence of activities, you can represent both serial and parallel activities in the diagram. Each activity should represent a node in the network, and you can use arrows to show relationships between activities.



PERT diagram

Step 4. Estimating the time necessary for each activity

What distinguishes PERT from other techniques is its ability to deal with uncertainty in activity completion time. There are three-time estimates this model typically uses for each activity:

- **Optimistic time** - the shortest time in which the activity can be completed

- **The most likely time** - the completion time that has the highest probability
- **Pessimistic time** - the longest time in which the activity can be completed

After you identified the time estimates, you can calculate the expected time for each activity by using the following weighted average:

Expected time = (Optimistic + 4 x Most likely + Pessimistic) / 6

For example, imagine you are building a cottage. Drilling and planting the posts has an optimistic duration of 7 hours, an expected duration of 10 hours, and a pessimistic duration of 12 hours. The optimistic duration is counted once, the most likely time is counted four times and the pessimistic time is counted once.

The entire sum is then divided by 6 and the weighted average is 9,63.

You perform this calculation for each task and then insert it into duration column.

Optimistic	7h	x1
Most likely	10h	x4
Pessimistic	12h	x1
Expected	9.63h	$\left(\frac{7+40+12}{6}\right)$

PERT time estimation

Step 5. Identifying the critical path

By adding the times for the activities and determining the longest path, you create a critical path. The critical path involves the total amount of time necessary to complete the project. The total project time doesn't change if activities outside the critical path speed up or slow down.

PERT advantages

At its core, PERT gives you the ability to control complex and ambitious projects whose objectives can be highly critical in nature. It helps you determine the fastest possible route to complete the projects.

PERT gives you a number of benefits:

- In-depth analysis of project activities by viewing the activities both independently and in connection with each other, you get a clear view of the time and the budget required to finish the entire projects.
- What-if analysis helps you identify all the possibilities and uncertainties related to the project. By trying different combinations and choosing the most useful possibility, you eliminate the risk of having project surprises. Also, it helps you highlight the activities that require careful monitoring.

PERT helps project managers identify responsible departments and delegate roles to their team workers. By gathering information from multiple sources, you can easily coordinate

project activities and leverage communication between departments. Through efficient planning and decision-making, you encourage your team to truly invest their time and energy into delivering the best results.

PERT disadvantages

Even though PERT has proven to be effective in terms of reducing the expected project completion time, there are still some limitations we need to be aware of:

- Although PERT makes you clearly define all the activities on a project, it's sometimes impossible to predict every step. Changes happen during the project and they can seriously affect the initial PERT. While it's possible to make modifications, it takes a lot of time and energy without contributing much to the project.
- Project managers make time estimates and since they heavily depend on judgment, the numbers are only guesses, especially if the project manager has little experience with the activities at hand.

Overall, PERT allows you to have an idea of possible time variation and helps you assess the importance of problems you have to face along the way. Unlike most methods, PERT gives you the flexibility to identify the best-case and the worst-case scenarios and develop a strategy on how to best coordinate large-scale projects.

Critical Chain Project Management (CCPM)

Critical Chain Project Management is a schedule network analysis technique that takes into account task dependencies, limited resources availability (people, equipment, physical space), and buffers necessary to successfully complete the project. CCPM allows a project manager to plan and manage project's schedule by concentrating on resources used in Critical Path (also known as the Critical Chain).

The origin of Critical Chain Project Management

Critical Chain Project Management was developed by Dr. Eliyahu M. Goldratt in 1997 as a response to inability to complete tasks on time and within budget. Dr. Goldratt first introduced the method in his work on Theory of Constraints (TOC), which focuses on identifying and fixing bottlenecks in order to improve the entire workflow.

Why we need CCPM

Traditional project management is based on predictable experience and predictable tools. But, as much as we would like to be able to control every single aspect on a project, we can't. In traditional project management, a lot of time is spent on making accurate time estimates and reduce risks, which delays projects.

Critical Path Method isn't very helpful either because it's based on the idea that all the resources will be available at any given time, which is not always the case. Also, according to Critical Path Method, if any activity is delayed, the delay will pass on to the next activity and delay the entire project.

Unlike other techniques, Critical Chain Optimization (a huge part of CCPM) helps us determine which time estimates we can shorten and it allows us to evenly distribute workload thanks to flexible start times. Finally, Critical Chain Optimization ensures we don't need safety margin and all tasks are completed ahead of schedule.

What is CCPM

Critical Chain is the longest path in the network diagram that takes into consideration task dependency and resource availability. It's a modified form of Critical Path Method where project activities use aggressive time and have access to unlimited resources.

While Critical Path Method uses Float, CCPM uses buffers as strategic points that eliminate uncertainty around projects.

There are four types of buffers:

- **Project Buffer** - protects the project from missing its scheduled end date and keeps thecompletion date unchanged. It is inserted at the end of the project network diagram, between the last task and the completion date. It

protects project completion date, which might vary due to the changes in activity durations in the critical chain. In other words, the size of the project buffer depends on the activities in the critical chain.

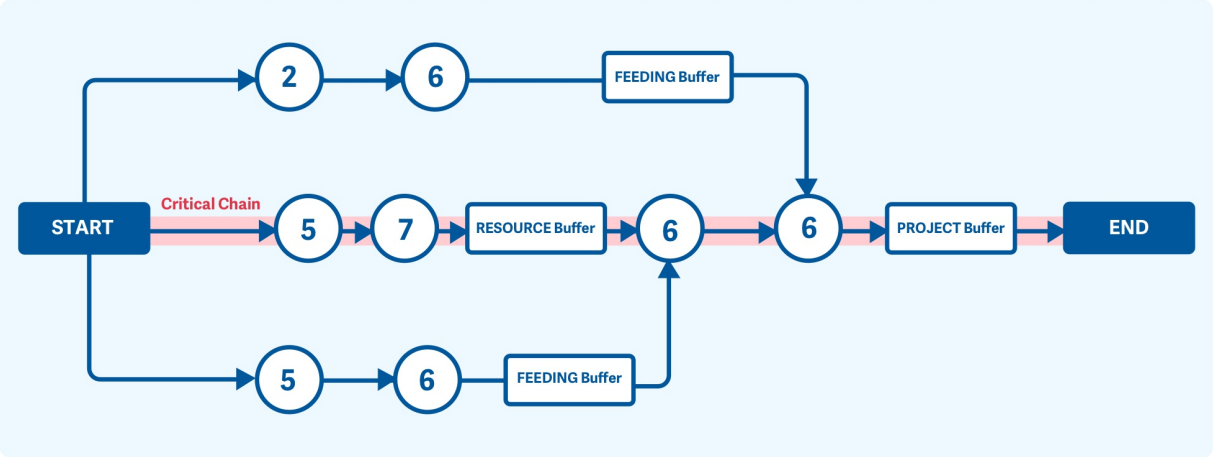
- **Feeding Buffer** - is inserted between the last task on a non-critical chain and thecritical chain. These buffers are typically added to a non-critical chain so that any delays on a non-critical chain don't affect the critical chain.
- **Resource Buffer** - these are set on the Critical Chain to ensure appropriate resources (, equipment) are available throughout the project when needed. These resources are commonly known as Critical Resources.
- **Capacity Buffer** - it sets on-call resources necessary in case unforeseen budget issues arise.

The specific steps in CCPM

There 9 steps you need to do in the CCPM process:

- Reduce all time estimates by 50% (the protection that is cut is inserted as a buffer in the project)
- Level the project plan and remove resource retentions (in this phase, a critical path is transformed into critical chain).
- Add a portion of reduced task estimates into the Project Buffer and insert it at the end of the project
- Insert Feeding Buffers at points where non-critical chain path meets critical chain path

- Protect the Critical Chain from unavailable resources by placing Resource Buffers where appropriate
- Insert Capacity Buffers where needed
- Schedule tasks as late as possible (this will help you prevent multitasking)
- Encourage aggressive task completion time and emphasize the importance of start time to complete the tasks as quickly as possible
- Manage buffers and gain information necessary for controlling the plan and taking recovery actions if needed



Critical Chain diagram

Here's a real world example of CCPM in action. Let's say you're building something. You created a plan and built a schedule based on the critical path. Although you have given enough thought to each project stage and started working on it, you suddenly realize there isn't enough equipment nor people to complete the project on time.

To prevent the project from failing, you consider resource

allocation into your project plan and modify the critical path. Essentially, you change the critical path into a critical chain and this way you create a much more realistic schedule.

What are the benefits of CCPM

Although CCPM is mainly used in multi-project environments that require a lot of resources, you can apply it on your projects regardless of whether you are running a small company or a big corporation.

CCPM is probably one of the most practical and the most important project management technique because of a number of benefits it offers:

- It allows people to become more focused on their tasks, thus increasing team productivity and efficiency
- It helps your team overcome the Student Syndrome phenomena (when people start working more as the deadline starts approaching)
- It avoids mismanagement of floats
- It considers the minimal time needed to complete the project
- It accelerates project completion
- It make significant reduction in capital requirements

Adaptive Project Framework (APF)

Adaptive project management is a structured and systematic process that allows you to gradually improve your decisions and practices, by learning from outcomes of the decisions that you took at previous stages in the project. As the name suggests, project management process changes and adapts to the needs of the organization, ultimately boosting business value.

Adaptive Project Framework (APF) has a number of variations, with adjusting scope at each iteration. Robert K. Wysocki, the author of Adaptive Project Framework, says that APF is more like creating a recipe than following one. He also explains that project managers are in charge of the approach, meaning they need to understand the situation and adapt their approach and techniques.

Main characteristics of Adaptive Framework are:

- Thrive on change
- Learn from discovery
- Client driven

What distinguishes APF from other methodologies is the fact that it places the client as the central figure who decides on project's next step. Also, the client has the power to completely change project direction.

Why do we need Adaptive Project Framework?

Traditional projects have a clear structure and a static strategy where project managers distribute tasks and keep the entire team under control. But the fast-paced technological advancements and ever-increasing demands of today's market have changed project management in three key areas:

- **Strategy** - it became more dynamic and difficult to predict
- **Work** - the development of new technologies speed up the pace of work
- **People** - they work collaboratively and strive towards creating a real team culture

Taking all these changes into the account, it's obvious that software development projects evolve as they go and that applying traditional project management would be futile.

Wysocki uses the analogy of a chef and a cook to explain the difference between the traditional and the adaptive project management: while a cook follows a recipe to the letter, a chef doesn't only have the knowledge and experience to create a recipe but also the ability to improvise if something unexpected happens.

How was the idea of APF born?

During the 1970's, two ecologists C.S. Holling and Carl J. Walters

researched how to predict the fish stocks, which are dependent on a number of factors that can't be controlled. As a result, they came up with adaptive project management, a method whose idea is "learning by doing".

APM has become an integral part of project management methodologies, and most big corporations and environmental organizations are using it on their large-scale projects. It has become top methodology of environmental engineering and is used on some of the biggest projects worldwide (such as Everglades and Grand Canyon National Park).

How does APF work?

Adaptive project management processes (planning, execution, monitoring, control, and evaluation) can be:

Passive

Passive Adaptive Management allows you to apply important lessons that you've learned during the process to the current management approach. The information you obtain from one iteration you can then use on the following one. In this way, you can minimize all the uncertainties related.

Active

Active Adaptive Management enables you to determine the best management strategy by learning through experiments.

Here are the essential steps you need to take when implementing Active Adaptive Management:

- Define a project strategy and make it flexible. This way, you'll be able to reverse decisions if necessary.
- Break your project plan into multiple iterations (phases).
- Create a more detailed plan that will include a schedule and a list of risks for the next phase. Make sure you don't make detailed plans for future iterations as you cannot predict all potential changes to the project.
- Perform a quantitative risk analysis, which should help you determine what will happen with project schedule if a certain risks occur.
- Execute one or more project scenarios and measure their results.
- Perform quantitative risk analysis again to gain a better insight into which risks actually occur and which ones don't.

Traditional project management vs Adaptive Project Framework

Future uncertainty prevents us from relying on traditional management, especially if we are involved in complex projects (eg. IT projects). Robert K Wysocki claims that traditional project methods are not applicable in such cases.

By using a traditional project management, you can achieve

positive results only if you have a clearly defined goal and a reasonable solution. However, If your goals are not specific and may change as project progress, you should consider using adaptive project management.

Agile project management vs Adaptive Project Framework

Agile project management was introduced to reduce complexity by breaking down ambitious and complex software development processes into smaller segments so you can make product changes even at later stages.

While Agile project management is mainly focused on the project organization, Adaptive PM takes it to another lever. Besides helping you organize your project more efficiently, it also gives you the ability to include quantitative analysis methods, which can help you:

- Analyze model and test hypothesis
- Measure the actual performance of the project
- Analyze project risk

The core values of APF

Since the client is in the center of attention in APF, they are given an opportunity to control the direction of the entire project. According to Wysocki, this way of thinking is embodied in six core

values of AFP:

- **Client focused** - it helps you stay focused on your client's needs as long as they are within the scope of ethical business practices
- **Client-driven** - It allows you to include the client in your project and keep them meaningfully involved by having project co-managers
- **Incremental results early and often** - ADP enables you to deliver a workable solution to the client as early as possible and keep them meaningfully engaged in the project
- **Continuous Questioning and Introspection** - refers to an openness and honesty that must exist between client team and development team in order to make the best decisions possible and deliver positive results
- **Change is the progress to a better solution** - by working with deliverables from the early stages of the project, both the client team and the development team will get the big picture of what else can be done to improve results.
- **Don't speculate on the future** - APF strips out all non-value-added work. Although the APF team will always be tempted to achieve perfection, they must resist the temptation. It's essential that your team doesn't waste their time and money on guessing but instead focus on the work your client will benefit from.

Extreme Project Management (XPM)

> *"Extreme project management is the art and science of facilitating and managing the flow of thoughts, emotions, and interactions in a way that produces valued outcomes under turbulent and complex conditions: those that feature high speed, high change, high uncertainty, and high stress." - Doug DeCarlo, author of eXtreme Project Management*

From development of new technologies and shift in customer needs, to economic conditions or some new groundbreaking ideas, a number of project requirements can change every day due to a variety of circumstances. This is where extreme project management enters the game.

Extreme projects are carried out in turbulent environments where it’s difficult to estimate the speed of the project and obstacles you will encounter. On extreme projects, things are unpredictable, planning is chaotic and just-in-time, and the entire project development is messy.

Despite their extreme nature, extreme projects can still contribute to success and extreme project management allows you to manage the unknown and unpredictable by self-correcting along the way.

Traditional vs Extreme Project Management

Traditional project management is a perfect solution for managing engineering and construction projects because they have a specific goal and a well-defined path on how to get there. But today, many projects don't have a proven path and a predictable life cycle, and requirements are constantly changing.

Unlike traditional management, where circumstances are highly predictable, extreme project management thrives in the chaotic environment where the level of certainty is very low. Also, traditional approach is more streamlined while extreme one is more flexible.

Traditional Project Management	Extreme Project Management
Manages the known	Manages the unknown
Slow and stable	Chaotic, messy and unpredictable
Aimed at producing the planned result	Self-correcting and making in-the-spot decisions
Focused on efficiency	Focused on effectiveness

Differences between traditional and extreme project management

Doug DeCarlo, the author of Extreme Project Management, points out the basic differences between the two approaches:

- Traditional project management is past oriented. Extreme project management is future oriented.
- Traditional project management makes people the servants of the process. Extreme project management makes the process the servant of people.
- Traditional project management is about centralizing control of people, processes, and tools. Extreme project

management is about distributing control.

- Traditional project management tries to take charge of the world (things, people, schedule). Extreme project management is about taking charge of yourself, your attitudes, and your approach to the world.
- Traditional project management is about managing. Extreme project management is about leading.

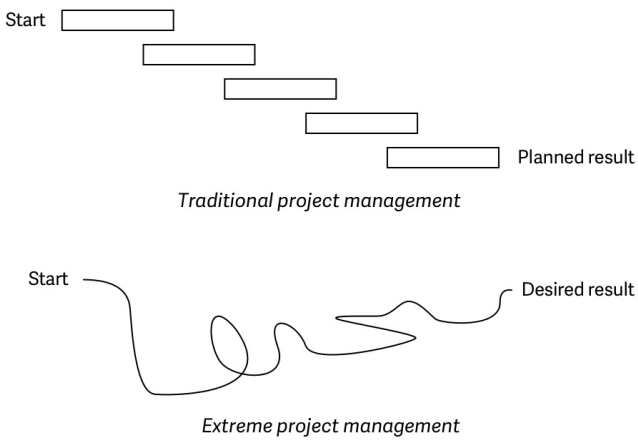
Extreme Project Management characteristics

DeCarlo compares extreme project management to jazz music. Although jazz may sound random and chaotic, it actually has its own framework which allows jazz musicians to improvise within it and make outstanding musical pieces.

- There is a rough idea about the goal but little idea about how to achieve that goal.
- All the standard tools, templates, and processes engineers used to apply in the past don’t make much contribution to the extreme project management.
- Instead of following the safe path, in extreme management project managers discuss the best alternative with the client, carry out the experiment, learn from what happens, and use their knowledge for the next project cycle.

The mindset as an important factor

It’s obvious that steps you need to take in extreme approach differ significantly from the steps in the traditional approach. The extreme approach requires a particular mindset, that is, a set of beliefs and assumptions of how things function. With this in mind, changing the mindset of your project team and adapting it to extreme circumstances they have to work in is the imperative.



Traditional vs Extreme project management

There are a few ground rules and expectations your project team has to adopt to successfully implement extreme approach:

- Requirements and project activities are chaotic and unpredictable
- Team should rely on uncertainty
- It’s virtually impossible to fully control this kind of projects
- Change is inevitable
- Flexibility and openness bring the feeling of security

Five steps of XPM life cycle model

Brian Vernham, the author of Agile Project Management for Government, suggests that there are five steps every extreme project management team needs to follow to carry out the project successfully:

- **Envision** - define your vision clearly before embarking on extreme project management.
- **Speculate** - have your team participate in creative thinking process and brainstorm ideas that will achieve your vision.
- **Innovate** - make your team test their speculations by coming up with innovative solutions.
- **Re-evaluate** - as the cycle approaches its end, your team must re-evaluate their work.
- **Disseminate** - after going through a learning process, it's essential to spread the knowledge and apply it to future stages of the project as well as future projects in general.

When to use XPM

Whether your team will employ a straightforward and well-structured traditional project management or the radical extreme project management approach depends on the project they are involved in. You should use extreme project management when your projects require:

- Fast-paced work
- Frequent changes as the project progresses due to the

dynamic environment

- A trial-and-error approach to see what works
- Self-correcting processes when things go wrong
- People-driven processes instead of process-driven (when people are in control of processes rather than the other way around)

Advantages of XPM

Unlike other methodologies that rely on software tools and templates, extreme approach is much more people-centric:

- **It's holistic** - although it includes methods, tools, and templates, they only make sense if they refer to the project as the whole. In other words, it allows you to view the project as a single system without analyzing its parts
- **It's people-centric** - it puts emphasis on project dynamics, meaning it allows stakeholders to communicate and interact. This helps you reach meaningful solutions and meet your client's needs.
- **It's humanistic** - one of the principles of this approach is takes into account quality of life of the stakeholder as they are baked into the project. Because people are an integral part of the project, their job satisfaction and the team culture they develop can have a profound effect on the business
- **It's business focused** - once you have reached the project's outcome, you can have a clear insight into how the project

can benefit your client. The team is constantly focused on delivering value early and often.

- **It's reality based** - it allows you to work in the highly unpredictable environment that is prone to change and helps you recognize that you cannot change the reality to adapt to your project.

Six Sigma

Six Sigma is a problem-solving methodology aimed at improving business and organizational performance by using a number of methods and tools. It primarily identifies and corrects errors and defect in manufacturing and business processes.

History of Six Sigma

Although Six Sigma was originally introduced in the 19th century with the bell curve by Carl Fredrick Grauss, it turned out to be inadequate and needed some corrections. It was brought back to life back in the mid-1980's when Motorola was facing a crisis: their product had poor quality, which affected their competitive advantage as they were struggling to beat the competition and secure their place on the market.

Motorola came up with an idea to develop Six Sigma Management System and reduce the variations in manufacturing processes that were causing defects. The results were stunning.

The level of quality of certain Motorola products experienced a tremendous increase, which encouraged other big corporations to follow the same path. Since then, many worldwide companies have adopted the Six Sigma approach with all its tools and techniques, and have saved hundreds of billions of dollars.

How Six Sigma works

Engineers at Motorola developed a scale that should evaluate the quality of processes based on their defects. At the top of the scale is Six Sigma, which is 99.9997% defect-free (or in other words, if you make a million car cylinders, there will only be 3.4 defects). This means that if your processes run at Six Sigma, they almost have no defects.

Sigma Level (Z)	Defects per Million Opportunities (DPMO)	Percent Defects (%)	Percent Success (Yield %)	Capability (CP)
1	691,462	69	31	0.33
2	308,538	31	69	0.67
3	66,807	6.7	93.3	1
4	6,210	0.62	99.38	1.33
5	233	0.023	99.977	1.67
6	3.4	0.00034	99.99966	2

Six Sigma scale

Six Sigma scale shows how well an important feature performs compared to its requirements. The lower the sigma level, the bigger the number of defects. Naturally, most businesses operate at 4 or 5 Sigma, which means that the losses that result from the poor quality cost them 10-15% of their revenue. The following table shows the universal Six Sigma scale.

Six Sigma concepts

Given its effectiveness and success, Six Sigma methodology is not a precise and narrowly defined term. Today, Six Sigma represents

a number of concepts:

- Six Sigma helps you solve the problems effectively and upscale your business and organization
- Six Sigma performance is a statistical term that refers to a process that produces fewer than 3.4 defects (errors) per millions of opportunities for defects.
- Six Sigma improvement is when the key outcomes of a business of work processes improve dramatically, by 70% or more
- Six Sigma deployment refers to the application of Six Sigma methodology across organization with all the assigned practices, roles, and procedures that match accepted standards

Six Sigma organization uses tools to:

- Boost performance
- Lower costs
- Grow revenue
- Improve customer satisfaction
- Increase capacity and capability
- Reduce complexity
- Lower cycle time
- Minimize defects and errors

DMAIC methodology

The Six Sigma methodology is defined by 5 DMAIC steps. DMAIC is a problem-solving technique that can help you handle any problems you meet along the way. It includes five main steps:

1. **Define** - you need to define the problem and clearly outlay your objectives. Your definition should include the exact problem with numeric representation and shouldn’t be vague.

2. **Measure** - once you define your problem, you need to decide what measurements to use to quantify the problem. You have to identify what part of the product is damaged, when they were damaged, and what's the level of the damage.

3. **Analyze** - after measuring, the next step is to collect and analyze data. At this point, it’s important to determine whether the error is valid or it happened at random, without any specific cause. You can use the collected data to compare it against measurements and evaluate its success.

4. **Improve** - the next stage includes developing solutions to the problems. Your team should create a test and launch pilot studies to find the most appropriate solution. After coming up with a solution, your team can start building a plan and developing a timeline.

5. **Control** - finally, in order to keep things under control and prevent recurrence, you need to perform control measurements monthly, daily, or yearly.

Six Sigma tools

To improve the quality of DMAIC model, there are a variety of tools, all of which can be divided into two categories:

- Process optimization tools that allow you and your team to create more efficient workflows.
- Statistical analysis tools that allow your team to analyze data more efficiently.

To be able to do calculations, you need to incorporate the tools into Six Sigma software, which will do the rest of the job for you.

Here are some of the essential tools that almost every Six Sigma Model uses:

Quality Function Deployment

It will help you identify customer requirements and rate them on a numerical scale. Next, you list various design options and list them based on their ability to address customer needs. Finally, the designs with the highest scores will become the solution that you should implement.

Fishbone Diagrams

It helps you identify which variables you should study further. First, you start with the specific problem and list all the variables in their respective categories that are affecting the problem. After listing all the variables, the expert team should determine which variables are most likely to be causing the problem.The diagram

looks like a fish skeleton, hence the name of the diagram.

Cause-and-effect Matrix

It helps your team identify, explore, and display all the possible causes and finally find the root cause to the problem.

Failure Modes and Effects Analysis

This tool allows you to focus on other processes and activities other than the issues that arise during the project development. This lets you list all possible failure scenarios, come up with solutions, and rank them according to how well they address your concerns. Finally, your team can prioritize things that could go wrong and develop necessary preventive measures.

Six Sigma belts

Since most large corporations and big industries incorporate Six Sigma into their business, the incorporation process requires not only ample time and energy but also huge financial resources.

At its core, we can incorporate Six Sigma in two ways:

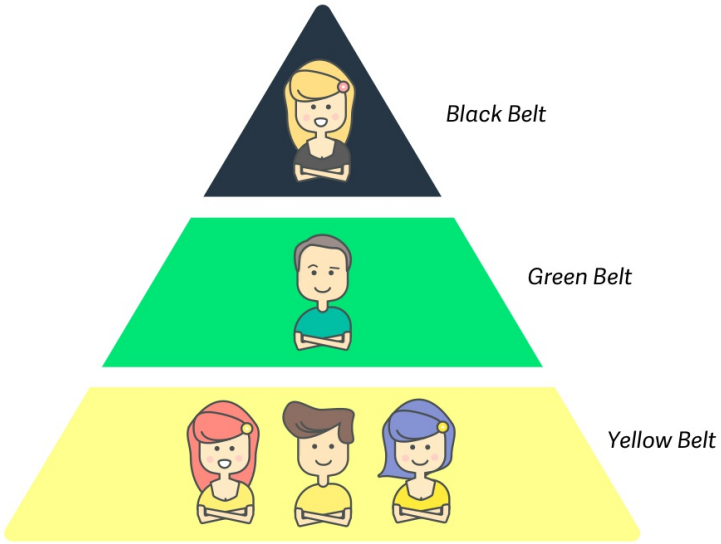
- Through a separate organization that provides services to the main business
- By making it an integral part of every employee's job

No matter what approach you choose, you need to keep in mind that Six Sigma focuses on the team and not on the individual. Each team has a certain level of certification that depends on the

degree of their expertise and responsibilities.

Teams are categorized as follows:

- **Black Belt** - represents professionals and experts who need to have a high degree of expertise and a wide knowledge of all Six Sigma tools and methods
- **Green Belt** - represents people who are in charge of solving a number of issues that arise in the manufacturing environment (if some issues become more complex, they can always consult the Black Belt). They usually handle administration and organization
- **Yellow Belt** - everyone else on the team. These employees don't know every project detail and therefore don't need to have extensive knowledge of Six Sigma. However, they are essential to the success as they help Green Belt people achieve their goals.



Belts in Six Sigma

Who uses Six Sigma

After the big Motorola success, many companies from manufacturing and transactional industries (banks and hospitals) to oil industry and even entertainment industry have incorporated six sigma methodology in their framework.

For example, while General Electric profited from $7 to $10 billion in 5 years, Bank of America saved hundreds of millions of dollars within the first 33 years of launching Six Sigma.

Six Sigma is inevitably a path to a dramatic improvement and brings immense value not only to the company but customers as well.

Theory of Constraints (ToC)

The Theory of Constraints is a methodology which helps you identify the crucial limiting factor (usually referred to as a constraint or a bottleneck), that stands in the way of achieving a goal. The main goal of theory of constraints is to improve that constraint to the point when it's no longer the limiting factor.

The goal and origin of Theory of Constraints

In his bestselling novel "The Goal" (1984), Dr. Eliyahu Goldratt developed the Theory of Constraints (TOC) and introduced it to a wide audience. Since then, TOC has continued to evolve, and today it is one of the most well known management techniques.

The Theory of Constraints hypotheses that every complex system, including manufacturing processes, consists of multiple linked activities, where one of them can disrupt the whole system (i.e. "weakest link in the chain").

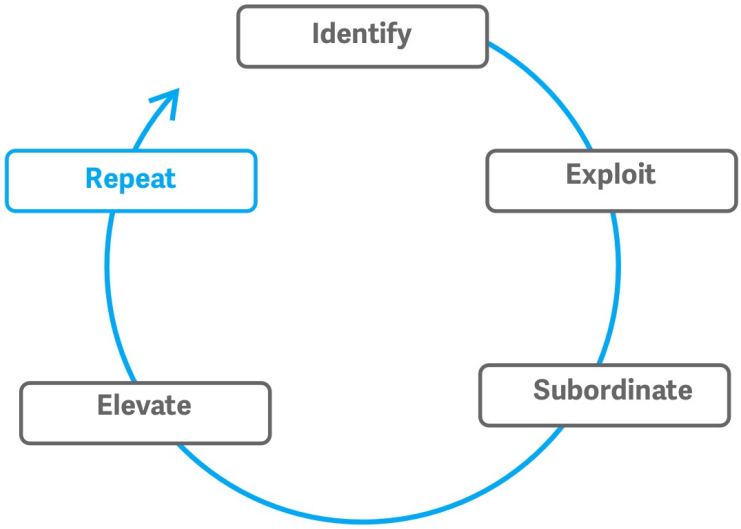
The goal of Theory of Constraints is to make a profit. Tools that will help you achieve that goal are:

- **The Five Focusing Steps** - a methodology that helps you identify and eliminate constraints.
- **The Thinking Processes** - tools that help you analyze and resolve problems.

- **Throughput Accounting** - a method that helps you measure performance and guide management decisions.

The five focusing steps

The five focusing steps are the cyclical process aimed at improving and eliminating bottlenecks you encounter on almost any project. They include the following steps:



The five focusing steps

- **Identify** - Identify the current constraint.
- **Exploit** - Make quick improvements by using existing resources.
- **Subordinate** - Review all the activities to ensure they support the needs of the constraint.
- **Elevate** - Take actions until the restraint has been eliminated. In some cases, this may require major

investments.

- **Repeat** - After resolving one constraint, repeat the cycle and work on improving new constraints that pop-up.

The thinking processes

The Thinking Processes are mainly used for complex systems with many activities that depend on one another. They are designed as scientific “cause and effect” tools. Their role is to identify root causes of undesirable effects (referred to as UDEs), and then remove the UDEs without developing new ones.

The Thinking Processes help you answer three vital questions:

- What do you need to change?
- What should you change it to?
- What actions will cause the change?

Throughput accounting

Unlike traditional accounting (where there is a strong emphasis on cutting expenses which are limited by reaching zero expenses), the Theory of Constraints focuses more on increasing the profit and so it doesn't have such limitations.

According to Theory of Constraints, inventory ties up cash that you can use more productively elsewhere. Overall, Theory of Constraints attempts to eliminate distortions that promote goals

contrary to the goal of increasing profit.

Throughput accounting has four measures:

- Net profit
- Return on investment
- Productivity
- Investment Turns

Essentially, there are three improvements you should be focusing on:

- Will throughput be increased?
- Will investment be reduced?
- Will Operating Expenses be reduced?

The benefits of ToC

One of the most beneficial characteristics of the Theory of Constraints is that it emphasizes the process of improvement.

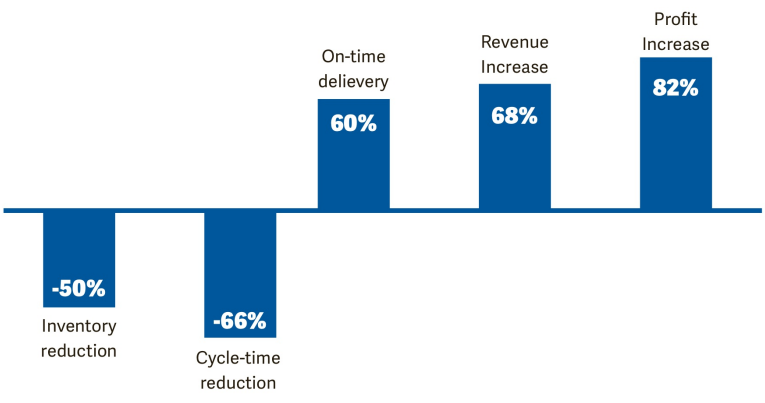
In environments where there is an urgent need to improve, TOC offers the best solution as it focuses primarily on creating rapid improvements.

A successful implementation of Theory of Constraints has a number of advantages:

- It improves capacity by optimizing the constraint
- It enables more product to be manufactured
- It increases profit
- It reduces lead time by optimizing the constraint
- It creates a smoother and faster product flow
- It lets you improve faster because you can focus all attention on one critical are, ie. the system constraint
- It reduces inventory and work-in-progress by eliminating bottlenecks

Theory of Constraints has had a tremendous impact on the business world and has produced remarkable results for companies worldwide.

Despite the popular belief that Goldratt' Theory of Constraints applies only to the manufacturing environment, it has had a major effect on other industries as well (including IT). It has also been developed for a wide variety of industries such as retail, banking, logistics, health care, and many other.



According to studies, companies achieved major success with Theory of Constraints

PRojects IN Controlled Environments (PRINCE2)

PRINCE2, which stands for PRojects IN Controlled Environments, is a flexible process-based method that can help you wrap up projects more effectively and efficiently. Ever since it was first introduced by UK government as the methodology for executing government projects, it has been gaining massive appeal worldwide.

PRINCE2 has become an integral part of every project development in both private and public sectors especially in countries like Germany, Spain, South Africa, Belgium and the United States.

The importance of PRINCE2

In the sea of project management methodologies, you might ask yourself: What can I achieve with PRINCE2? What makes it stand out from other methodologies? How can it help my business grow?

In its core, every project development, especially in IT world, is highly unpredictable in nature. PRINCE2 will help you address the common cause of the project failure, and order your project into defined steps to make it more organized, logical and easy to handle. In other words, it allows you to structure each stage of the project in details and tie up all the loose ends once the project

finishes.

PRINCE2 framework

We all know that poorly organized schedule and or lack of knowledge necessary to control all of the stages of project development can lead to irreversible consequences. For example, FBI's ambitious project to replace paper based reporting of crimes with an online system fell flat due to lack of sound communication and thorough planning.The project was scrapped in 2005 and it cost $170 M.

According to PRINCE2 framework, a project should have:

- an organized and controlled start
- an organized and controlled middle
- an organized and controlled end

7 basic principles of PRINCE2 methodology

To better understand how PRINCE2 works, you need to become familiar with the 7 basic principles, the building blocks of this widely popular methodology:

- **Continued justification of your business** - unless there is a justifiable reason to run and manage the project (a defined customer and realistic benefits) the project should be closed.

- **Learning from experience** - PRINCE2 will allow your team to take part in the learning process, and continually upgrade their knowledge
- **Clearly defined roles and responsibilities** - each member of the team should have a clear insight into what they should be focusing on and what their teams are responsible for.
- **Running the management in stages** - the project is divided into separate stages and which are planned, monitored and controlled with the aim to record lessons learned and confirm is on the right track.
- **Manage by exception** - people on the team have the right amount of authority to manage and coordinate the project within the controlled environment. A project manager is in charge of delegating tasks and controlling the budget, the cost and the deadlines. In case certain, detrimental issues arise, project board enters the game and decides what the next step should be.
- **Maintaining focus on the quality of products** - teams continually check the product delivery, definition and the quality requirements
- **The approach is adjusted for each project** - PRINCE2 method is tailored to suit the requirements of each project (environment, size, complexity, importance, capability, and risk)

7 basic themes of PRINCE2

To run the project management sufficiently, managing all of its stages is not enough. If you want to be able to control your project, you need to have a deep insight into all of its aspects and address them accordingly.

When it comes to PRINCE2 method, there are seven themes every project manager needs to tackle to run the project smoothly:

- **Business case**: create and maintain a record of business justification for the project
- **Organization**: define the roles and responsibilities of each member of your team
- **Quality**: what quality requirements and measures do you need to focus on and what steps do you need to take to deliver them
- **Plan**: what steps you need to take and what techniques you need to use to develop the plan of the project
- **Risk**: clearly identify the risks and opportunities that could affect the project
- **Change**: how you as a project manager will evaluate and tackle the changes on the project
- **Progress**: measure how fruitful the project performance is and whether the project is making any progress

7 processes of PRINCE2 methodology

Finally, “projects in controlled environments” is a process-driven project management method. There are seven processes you need to go through when managing the project:

- **Starting up the project** - assembling the project team, creating the project brief, agreeing upon project approach and approving the next phase of the project
- **Starting-Initiating the project** - documenting the project plan, risks, project controls and the plan for the next stage
- **Starting-Directing a project** - controlling the project, giving ad-hoc direction and confirming the end of the project
- **Starting-Controlling a stage** - controlling each stage separately by assessing the progress, managing issues, reviewing status and taking corrective actions
- **Starting-Managing stage boundaries** - updating the project plan and deciding what to do with stages once they have exceeded the tolerance level
- **Starting-Managing product delivery** - managing the

execution of the project work, while making sure that the deliverables live up to the expectationsn

- **Starting-Closing a project** - project wrap up including documentation and outcomes

The advantages of PRINCE2

- Easy to manage because it can be broken into several stages. This keeps your team's focus on delivering the positive outcome
- Improves the communication not only between team members but also between the team and other stakeholders which gives the team more control over the project
- It allows the team to save a lot of their time and become more economical when it comes to using other resources and cutting down on some major costs along the way
- The flexibility of this approach allows your team to make decisions and changes as the project develops, thus minimizing the impact of possible issues that may arise
- It allows your team to recognize the deviations and possible problems at early stages and have a clear view of the lessons learned which they can use on future projects

The disadvantages of PRINCE2

PRINCE2 may not be tailored for some modern project

management methodologies (such as projects in the software industry) because it does not offer the level of flexibility these methodologies usually offer.

PART III

Methodologies and Frameworks Used in Software Development

Waterfall

Waterfall is a project management approach where a project is completed in distinct stages and moved step by step toward ultimate release to consumers. You make a big plan upfront and then execute in a linear fashion, hoping there won’t be any changes in the plan.

When you take traditional project management and apply it to software development, you get Waterfall. As such, no one invented waterfall - instead, we gave it name once we realized that there are others ways to manage projects (like agile project management).

Waterfall was the first software development methodology, inherited from manufacturing and construction industry where you can't afford to iterate (after you've built a tower or a bridge you can't go back to "improve" the foundation). But because software is prone to frequent change, waterfall is not the best solution.

Waterfall is often mentioned alongside Agile and stands in contrast to it. The main difference between them is that waterfall doesn't react well to frequent changes, which is why it gets a bad reputation in software development community, where frequent changes are the norm.

Phases in waterfall projects

All tasks on waterfall projects are grouped by type of activity and each projects follows the same phases:

- **Requirements** - where we analyze business needs and document what software needs to do
- **Design** - where we choose the technology, create diagrams, and plan software architecture
- **Coding** - where we figure out how to solve problems and write code
- **Testing** - where we make sure the code does what it supposed to do without breaking anything
- **Operations** - where we deploy the code to production environment and provide support

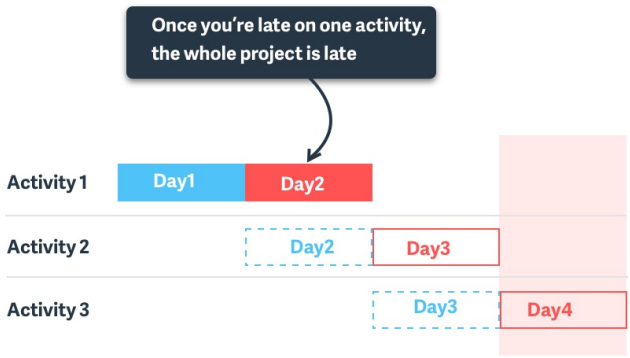
Once you put all the activities on a Gantt chart, you get something that looks like slopes of a waterfall, hence the name.

Usually 20–40% of the time is spent on requirements and design, 30–40% on coding, and the rest on testing and operations.

Activities on waterfall projects have to happen in the exact order and one set of activities can't start before the previous one ends. This is why planning is the most important thing on waterfall projects: if you don’t plan right, a phase will be late and will push

every other subsequent phase, thus putting the whole project over deadline.

The problem with using waterfall method on software project is that planning is very tricky in software development. You can never be 100% sure how much time you'll need on something or how much time you'll spend debugging. As a result, waterfall is risky.



Task interdependence on waterfall projects

Advantages of waterfall

Extensive documentation

Because you can't go back to a previous activity, you're forced to create a comprehensive documentation from the start, listing all the requirements you can think of.

Knowledge stays in the organization

When you have extensive documentation, knowledge won't get lost if someone leaves. Also, you don't have to spend time on training new members as they can familiarize with the project by

reading the documentation.

Team members can better plan their time

Because everyone knows in advance on what they'll work, they can be assigned on multiple projects at the same time.

Easy to understand

Waterfall projects are divided in discrete and easily understandable phases. As a result, project management is straightforward and the process is easily understandable even to non-developers.

Client knows what to expect

Clients can know in advance the cost and timeline of the project so they can plan their business activities and manage cash flow according to the plan.

Client input not required

After requirements phase, client input is minimal (save for occasional reviews, approvals, and status meetings). This means you don't have coordinate with them and wait for when they're available.

Easier to measure

Because waterfall projects are simple, it's much easier to measure your progress by quickly looking at a Gantt chart.

Better design

Products have a higher cohesion because during the design phase you know everything that must be taken into account. There is no one-feature-at-a-time problem that leads to usability problems down the road.

Disadvantages of waterfall

No going back

Once you're finished with one activity, it's difficult and expensive to go back and make changes. This puts a huge pressure on the planning.

No room for error during requirements phase

Everything relies heavily on the requirements phase and if you make an error, the project is doomed.

Deadline creep

Once one activity is late, all the other activities are late too, including the project deadline.

QA too late to be useful

Testing is done at the end of the project which means that developers can't improve how they write code based on QA feedback.

Bug ridden software

Because the testing is done at the end, most teams tend to rush the testing in order to deliver the project on time and hit their incentives. This short-term wins lead to sub-par quality and long-term problems.

Not what the client actually needs

Most of the time, clients can't articulate what they need until they see what they don't need. If the client realizes they need more than they initially thought, the project plan will need a major overhaul (as well as the budget).

Unexpected problems

Designers can't foresee all the problems that will arise from their design, and once those problems surface, it's very difficult to fix them.

On what types of project you should use waterfall

Waterfall is suited for projects where:

- budget, requirements, and scope are fixed (eg. you're building a one-off project which doesn't need further development)
- you can accurately estimate the work (you're familiar with technology and you've done the same work before)
- you can't afford to iterate (eg. you're building a heart rate monitoring software)

- project is innately low-risk (you're building a clone of something that already works)
- project has a hard ship date (eg. you have to ship a video game by Christmas)
- your users can't or won't update software (doesn't apply to web applications where updates are seamless)

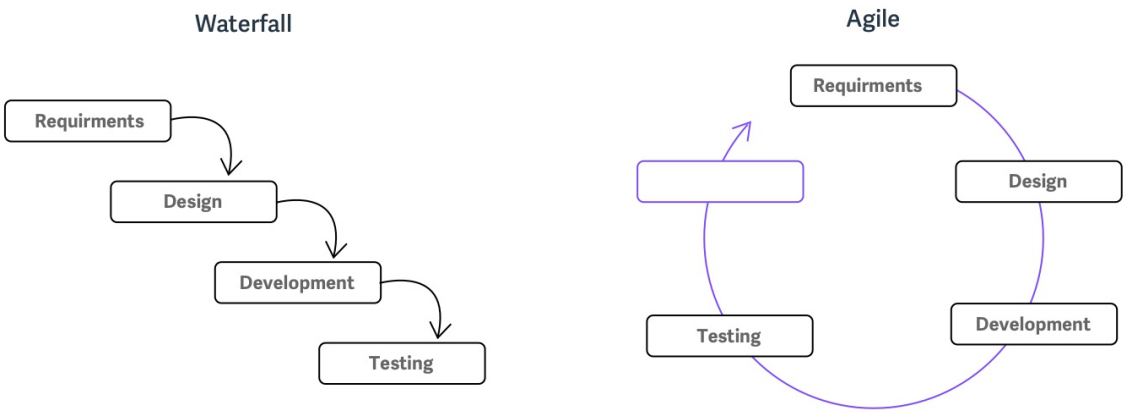
You shouldn’t use waterfall:

- where a working prototype is more important than quality (eg. you first need to test if there’s a market demand)
- when you don't know what the final product should look like
- where client doesn't know exactly what they need
- when the product is made for an industry with rapidly changing standards
- when you know you won't get the product first the right time and have to incorporate user feedback
- when your users are happy with v1.0 and you can ship additional features as time goes on

Whether you'll use agile or waterfall doesn't matter on your preference but type of project and your customer/client. While strictly speaking agile is better for software development (see the statistics here), if you can't iterate, you have to use waterfall.

Waterfall is not that different from agile

Waterfall is always mentioned as the antithesis to Agile, which makes sense. After all, waterfall projects have a hard time dealing with changes while agile projects welcome change. At least in theory.



Difference between waterfall and agile, in theory

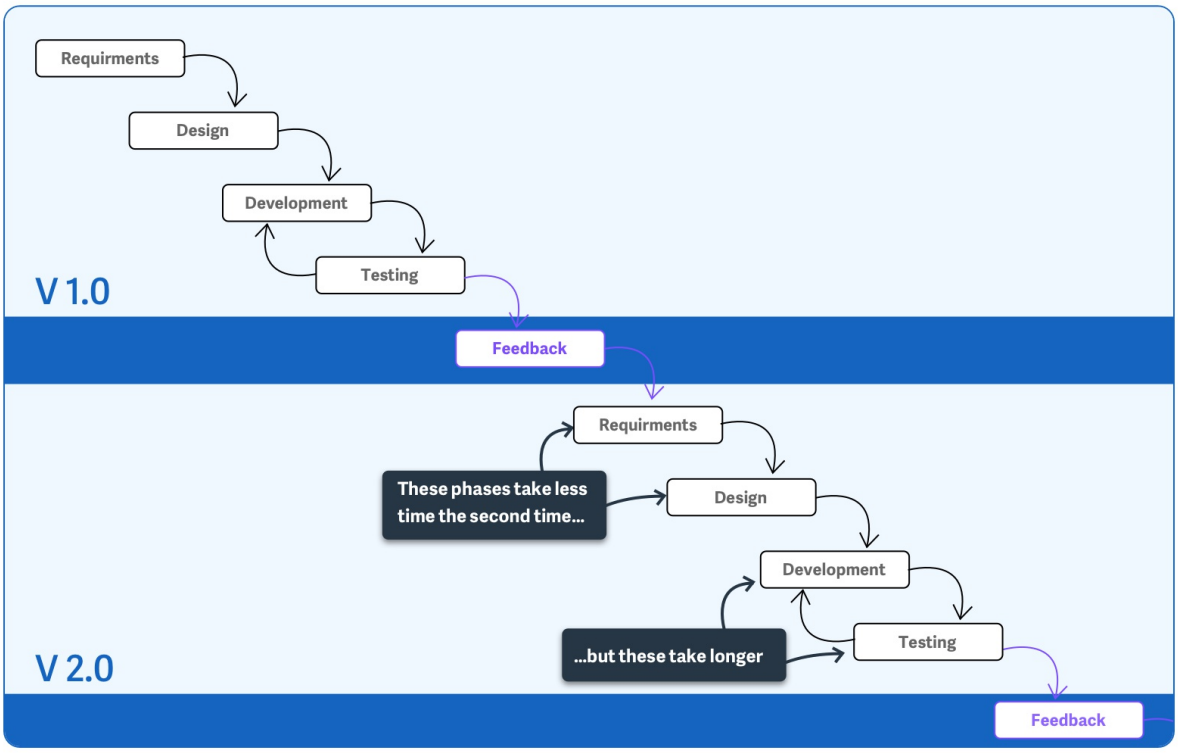
The truth is, no matter what methodology you use, change is not a good thing. Change always means additional scope, delay, and expenses. Agile is better at minimizing the effects of change, but they still happen. Also, agile teams have the culture where change is OK, which is maybe the most important benefit of being agile.

But once you scratch behind the surface and look both from purely process perspective, waterfall is very similar to agile.

Once you break down any agile workflow, you'll still get a set of activities that follow one another, which eerily resembles Waterfall. And if you treat waterfall projects as smaller phases within a big project, you'll end up with agile.

In other words, activities on a project are waterfall and if you treat

the whole project as a series of iterations, it's agile.



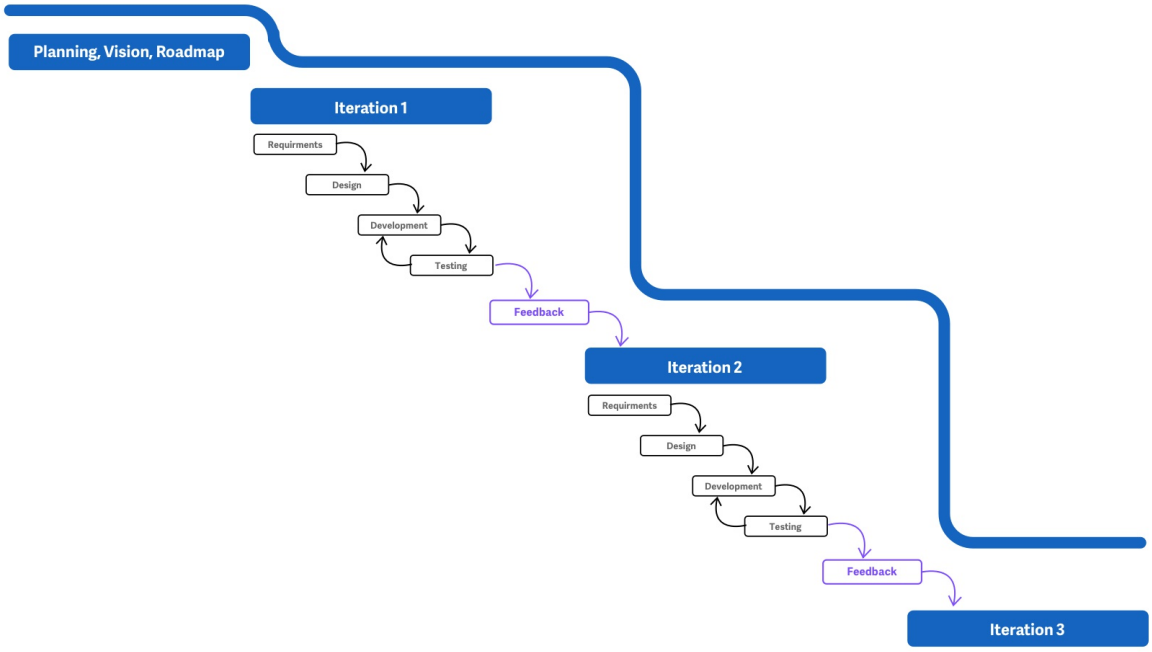
Agile = Waterfall + Iterations

Whether you're agile or waterfall ultimately depends on whether your client expects the first version to be bad. And waterfall projects are projects where the client decided on zero iterations.

> *In agile projects, the number of iterations is decided on by the customer. Because things are all done within an iteration in agile, the logical assumption was that an iteration equaled a project. But an iteration is more properly referred to as a phase or sub-phase of the project. - PMI*

As you can see, agile still fits in the traditional project management, only the point of view changes. Instead of treating each iteration as a separate project, iterations are just phases in one big project.

The real difference between the waterfall method and agile is that in waterfall the clients is heavily engaged at the beginning of the project and then their engagement declines; while in agile, the client is constantly engaged.



Agile projects follow the similar flow as waterfall projects

So what this all means in practice? It means no organization is purely agile or waterfall. Agile and waterfall are more about the culture and type of work the organization does than how they do it. You'll find that most organizations divide the project into waterfall milestone but work according to agile principles between those milestones.

> *A problem common with comparing agile and waterfall is the labeling. Few, if any, companies are purely "agile" or "waterfall". They are more*

mindsets that encompass a wide variety of practices and approaches to development. Labels are convenient for helping make an argument, often with cute little straw-man statements to help reinforce preconceived notions. - Clinton Keith

Agile

Agile is an approach to project management that favors responding to change over careful planning. Agile is not a methodology but a set of principles (as defined in Agile Manifesto in 2001) that suggests how we should approach project management.

Basically, there are two ways you can manage software development projects:

- **Waterfall**: plan everything in advance, then build according to the plan for the next whole year
- **Agile**: plan what you'll build in the next few weeks, and see how it goes from there

How to know whether you're agile? Just because you're using Kanban boards or Scrum doesn't automatically mean you're agile. To really be agile, you have to share the agile values, as defined in Agile Manifesto.

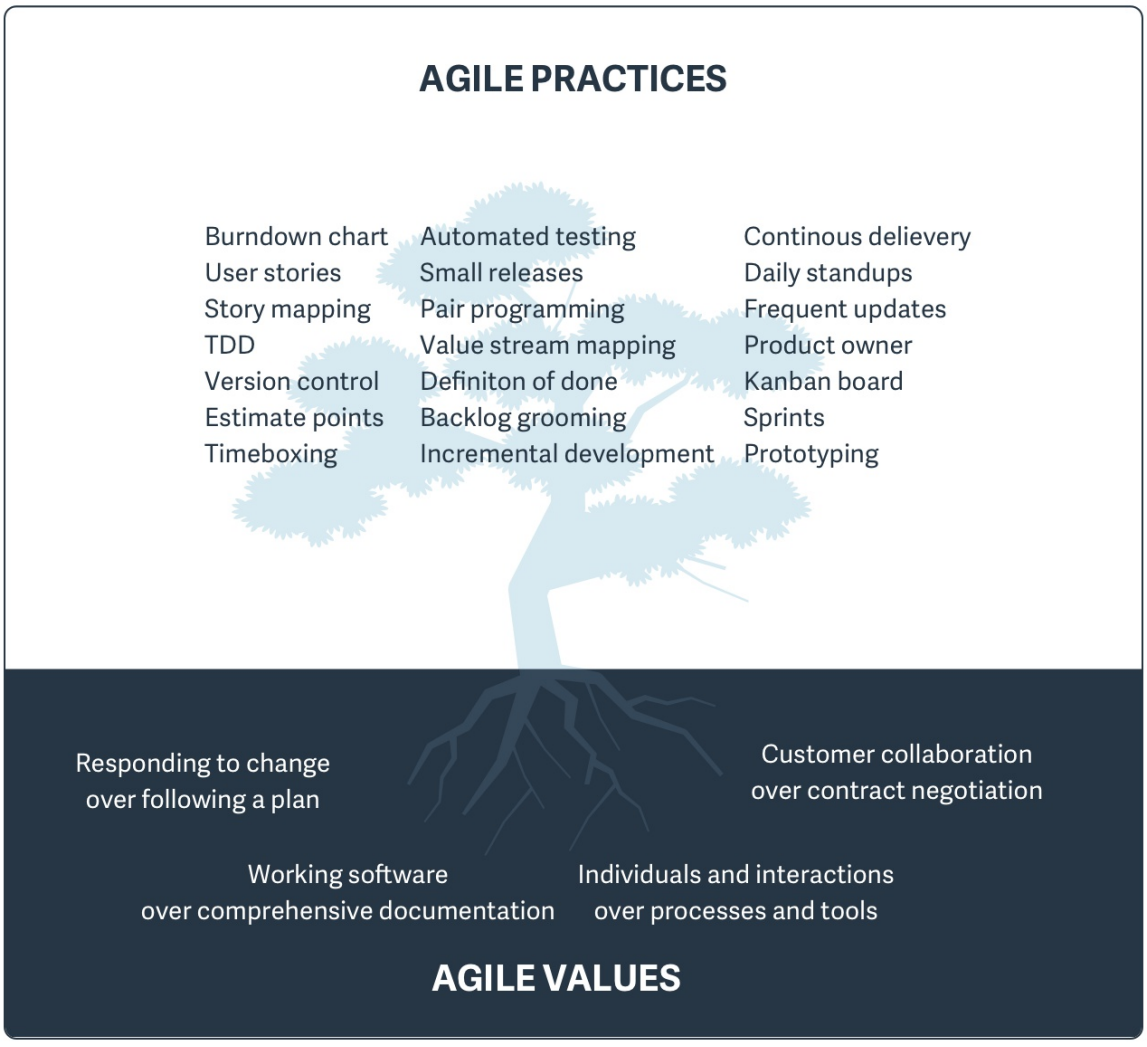
Agile Manifesto

In 2001, 17 software developers met in Utah to discuss their processes that were different from the usual waterfall project management approach. Together, they defined the concept of agile software development in Agile Manifesto. Today, when we say that some methodology is agile, it means it follow the value

and principles from Agile Manifesto.

The group recognized that there's no one-size-fits-all approach, so Agile Manifesto doesn't prescribe how to run projects. Instead, it defines the mindset on how to best manage software projects.

Most important part of Agile Manifesto are the 4 values. They are the heart of what it means to be agile.



Agile values help you focus on what's important. For example, one of the values is "working software over comprehensive

documentation". It doesn't mean that documentation is bad - it means that if you have to choose whether to spend your time on writing a detailed user story or fixing a bug, you should choose the latter.

Individuals and interactions over processes and tools

Knowledge workers prefer autonomy. So in software development it's more important to let people solve problems by collaborating than forcing them to follow a procedure for the sake of satisfying some dusty policy.

Every company needs processes (especially after they've grown to a certain size), but you must know why a rule is in place and when you should break it. For example, when daily stand-ups stop being useful, don't force them just because some agile methodology says you must have them.

The way you know when process doesn't work is when people can't collaborate efficiently anymore. People are the engine behind every project. If they can't interact because of hierarchy or a lengthy/complex protocol, they have to spend more time on managing tools and processes than doing their job.

> *Good process serves you so you can serve customers. But if you're not watchful, the process can become the thing. The process becomes the proxy for the result you want. You stop looking at outcomes and just make sure you're doing the process right. It's not that rare to hear a junior leader defend a bad outcome with something like, "Well, we*

followed the process." A more experienced leader will use it as an opportunity to investigate and improve the process. The process is not the thing. It's always worth asking, do we own the process or does the process own us? - Jeff Bezos

There's something really wrong with our definition of what a 'completed project' is. If it means 'Did Chris get all his project tasks done?' then it was a success. But if we wanted the project in production that fulfilled the business goals, without setting the entire business on fire, we should call it a total failure. - The Phoenix Project: A Novel About IT, DevOps, And Helping Your Business Win

Working software over comprehensive documentation

In traditional project management, phases happen in sequence and if you mess up the first phase (requirement gathering and documentation), every other phase will suffer. That's why waterfall needs comprehensive documentation. But on agile project we expect things to change.

Do you really want to spend your whole time updating the documentation? What matters the most is having a working product that real users can test. If you had to choose between fixing a bug and writing a report on it, fixing it is the best use of your time.

This doesn't mean that you should forsake documentation. Developers fall in this trap often and write terse one-line user stories, which creates trouble for QA and maintainers because

they can't figure out the proper user acceptance criteria.

The perfect documentation should be "Just Barely Good Enough". Too much and it goes to waste or can't be trusted because it's out of sync with code; Too little and it's difficult to maintain and get new team members up to speed.

When writing documentation, you should ask yourself what would you want to know if you joined the team tomorrow and document based on that. If you have trouble with documentation, grab a copy of Living Documentation by Cyrille Martraire.

Customer collaboration over contract negotiation

Contracts create the culture where change isn't an option. Agile creates the culture where change is expected. But how do you manage change? By collaborating with customers.

Agile presupposes that you have unlimited access to your customers and that you can always sit down with them and talk. Developers are natural problem solvers but they need access to the customer so they can better understand what the real problem is.

Contracts are useful, but they have a nasty side effect: people tend to care more about delivering the project within time and budget than fulfilling the real business goal. Further, when the team falls behind schedule, they are pressured to get things done which results in frustration, panic, and lower quality.

Also, when you sign a contract early in the lifecycle, you're guesstimating and more often than not, you're wrong. But you still try to hit the milestone even though they have nothing to do with real needs.

That's why agile favors customer collaboration and delivering work in small increments. This lets scope work as you gather more information and discover what you don't know.

Responding to change over following a plan

The more time you spend on planning, the more you resist changes lest your efforts go to waste. But the goal is not to deliver project according to the plan (within time and budget) - the real goal is to satisfy some business goal, and if it means completely changing your plan, then so must be it.

It's more important to build what you really need than to blindly follow an obsolete plan. Developers may hate it when they their code becomes invalidated, but clients hate it even more when they don't get the product they need.

That's why agile favors shorter lead time and encourages teams to chop things up in smaller deliverables so they won't have to redo large chunks of work. This means that you are never done with requirements gathering and design phases but you continually revisit them throughout the lifecycle.

Elements of agile

Culture where change is expected

Agile isn't about using Kanban boards, having daily stand-ups, or anything similar (those are elements of specific agile methodologies). Agile, at its core, is mindset where everyone, from employees to clients, expects change. You can't promise your client everything at once or a firm deadline because both you and the client know that's unrealistic. But you can promise them that you'll give them something they can use and listen.

Incremental development

Each iteration builds on previous work, making the product better gradually. Also, you don't wait for completed work to pile up before releasing it all at once - you release it as soon as it's finished. An iteration might not add enough features to warrant a marketing campaign, but that doesn't matter because the ultimate goal is to give customers value.

Frequent release

Because software is developed incrementally, you can have shorter cycles, where at the end of cycle you ship new features/updates. This way, customers can get value as soon as possible and validate it. If the work doesn't satisfy their needs, you can learn that before you spend more time on development.

Short feedback loop

Because releases are more frequent, you can get feedback faster. And because you can get feedback faster, you can more quickly

change the product and give value.

High level of client involvement

In order to reap the full benefits of short feedback loop and frequent releases, you need a high level of client involvement. You need to talk to your client after each cycle and see how they use the software and if it they derive value from it.

Sometimes the client is not available to give you feedback. Some agile methodologies (like Scrum) solve this problem by having a special role on the team called Product Owner. This person serves as a customer representative and act on behalf of the customer. If a developer has any question, they ask the product owner instead of the customer. Product owner also reviews progress and re-evaluates priorities at the end of each iteration.

Elements of agile project management

Advantages and disadvantages of agile

Today, agile is such a buzzword that teams outside software development try to incorporate it into their workflow. But agile is not for everyone.

For example, a marketing agency can never implement agile because clients don't want to pay for a half-finished marketing campaign and iterate. There are revisions, but their number is clearly specified in the contract. Plus, there are no such thing as a "working increments" - you either have the deliverables or you don't.

Agile isn't the right approach for every software project either. If you don't have access to customers, can't iterate, or if you have complex organizational structure, it's very difficult to adhere to agile principles.

Agile works best when

- You can't estimate the time you'll need and don't know the full scope of requirements
- You don't know whether there's a need on the market for your software
- You can't map out the business needs so the design needs to emerge through trial and error
- You have unlimited access to your customer who's ready for extensive involvement
- You can afford to iterate and don't need to deliver a fully functional software at once
- Neither you nor your client have a complex bureaucracy that delays decision
- Clients don't have a fixed budget/schedule

- You need to capture the market before there's any competition
- Your customers don't have trouble updating their software (or don't even notice it, eg. they use a web app)

As you can see, agile is more suited for small-to-medium size organizations than corporations. The reason is simple: the less people there are, the easier it is to make a decision and respond to change. Also, agile is more suited for product companies over consultancies.

Agile is also great for startups, where "fail fast" is the dominant mantra. Venture capitalists encourage startups to try crazy ideas and let the markets do the work. Most of the ideas will fail those few that succeed will change the world.

Advantages of agile

- You can deploy software quicker so your customer can get value sooner rather than later
- You waste less resources because you always work on up-to-date tasks
- You can better adapt to change and respond faster
- Faster turnaround times
- You can detect and fix issues and defects faster
- You spend less time on bureaucracy and meaningless work
- There's a big community of agile practitioners with whom

you can share knowledge

- You can get immediate feedback (which also improves team morale)
- Developers can improve their coding skills based on QA feedback
- You don't have to worry about premature optimization
- You can experiment and test ideas because it costs are low

Disadvantages of agile

- Documentation tends to get sidetracked, which makes it harder for new members to get up to speed
- It's more difficult to measure progress than in waterfall because progress happens across several cycles
- Agile demands more time and energy from everyone because developers and customers must constantly interact with each other
- When developers run out of work, they can't work on a different project because they'll be needed soon
- Projects can become ever-lasting because there's no clear end
- Scope creep and experience rot
- Clients who work on a specified budget or schedule can't know how much the project will actually cost, which makes for a very complicated sales cycle (until iteration ends is not something clients like to hear)

- Product lacks overall design, both from UX and architecture point of view, which leads to problems the more you work on the product.
- Teams can get sidetracked into delivering new functionality at the expense of technical debt, which increases the amount of unplanned work
- Features that are too big to fit into one or even several cycles are avoided because they don't fit in nicely into the philosophy
- You need a long term vision for the product and actively work on communicating it
- Products lack cohesion and the user journey is fragmented because the design is fragmented. The more time passes, the more disjointed software ends up.
- Short cycles don't leave enough time for the design thinking process so designers have to redevelop the experience over and over due to negative feedback.
- Check here for some more Scrum sprint planning anti-patterns and product backlog and refinement anti-patterns

Workers in tech don’t usually feel like they have the ability to focus on craft — especially when it comes to visual design. When you're constantly iterating, constantly pushing new versions out, you can’t invest time in seemingly unnecessary details that will be lost in tomorrow’s update. - Jessica Hische

Kanban

Kanban is a Japanese term that translates roughly as “ billboard”, is a method that helps organizations visualize, plan and organize their work by controlling various logistic elements and prioritizing tasks on boards.

As a result, Kanban helps teams eliminate bottlenecks and achieve outstanding quality improvements. To put it simply, Kanban offers a project manager and his team the place where they can manage their tasks, lists and files. This boosts the team’s productivity and helps them complete their projects more efficiently.

One of the most quoted sayings in Kanban is: “Stop starting and start finishing”. With so many tasks at hand, this seems like a mission impossible. However, with Kanban’s philosophy, you will be able to create a perfect scheduling system that will tell you what to produce, when to produce it, and how much to produce.

There’s no doubt that Kanban has entered a variety of business areas and is becoming all the rage. In today’s business world Kanban is known as one of the clearest, simplest and the most effective tools for project management.

First implementation of Kanban

Back in 1940’s, a Japanese Toyota engineer Taiichi Ohno was fascinated by the system American grocery stores used to stock their shelves. They stocked only as many items as that day's customers needed.

This inspired him to apply the method in Toyota production system and deliver the same level of efficiency on their assembly line. Basically, their goal was to keep their inventory low but always adequate to meet the demand for the parts. They named it Kanban (Khan - Ban), also known as “just in time” system.

Although it was initially gaining popularity in manufacturing industries, it has undergone some changes over the last few years and has been modified by David Anderson. With the rise of IT world and the digital age we are all part of, Kanban method is usually presented online as visual panels with virtual sticky notes which you can move around to organize tasks and to-do items.

Six core practices of Kanban

After using Kanban system at Microsoft in the mid-2000s, David J. Anderson was so thrilled about the positive results that he decided to use the method as the essential ingredient of future continuous improvement of company’s workflow.

In his book Kanban - Successful evolutionary change for your technology business, he focused on five practices that lead to such positive outcomes:

1. **Visualize your work** - use the board to represent your workflow, columns to represent the steps in this process, and list the relevant tasks of work items.

2. **Limit work in Process (WIP)** - be more effective and accomplish more by doing less. Delegate only ideal number of tasks to your team. Don't pull more work until your team completes something.

3. **Manage Flow** - start where you are, with the existing process, review where inefficiencies lie and improve continuously. Learn from your mistakes and prevent bottlenecks from arising again.

4. **Make management policies explicit** - define the whole process and make sure everyone understands how things work and what your goals are. For example, define the criteria that each step of the process has to meet to called “done”.

5. **Improve collaboratively** - when your team clearly understands the theories about work, processes and workflow, they are more likely to exchange their ideas and suggest improvement actions.

Four basic principles in Kanban

Kanban project management is a philosophy which suggests that you can gradually improve whatever you are working on and implement continuous, incremental and evolutionary changes. To achieve this goal, you need to follow the four basic principles that

underlie this revolutionary method:

Start with what you do now

The good thing about Kanban is that you don't need to make big changes before you implement this methodology. You can simply overlay Kanban properties on your existing workflow, address issues and make some important changes over time.

Pursue evolutionary change

Encourage small but evolutionary changes to processes and run project management that will meet minimal resistance.

Respect the current roles and responsibilities

While you may be satisfied with how certain elements and processes are working, you also need to seek out the way how to drive out fear to be able to make necessary future changes. By agreeing to respect the current roles, responsibilities, and process, you will gain broader support for Kanban initiative. This will help you implement Kanban method more easily.

Encourage acts of leadership at all levels

You don't need to be a team leader or an executive to encourage continual improvement and reach optimal results. With Kanban, some of the best leadership comes from everyday acts by common people who are a part of a team.

Kanban board

A Kanban board is a field on which Kanban Cards represent the individual tasks in progress which are categorized according to priority and delivery. Simply put, Kanban is an excellent way to keep track of your team's workflow. Today, Kanban boards are mostly used in Agile or Lean software development teams in the form of online collaboration tools aimed at boosting team's productivity and stimulating inspiring ideas.

A basic Kanban board organisation

The integral part of Kanban board is Kanban cards. Each card represents individual work item or a task, and each consists of important data for that specific task. Cards are placed on the board in a visual way in order to show the current stage of the task and they are usually color-coded to show what type of task they are.

Kanban board consists of three sections:

- Waiting
- Work in progress

- Completed work

After the tasks have been delegated, the team will take the cards and move them across the sections as they complete their tasks.

Keep in mind that Kanban board can be used in many forms and that today teams use it in different ways to share ideas and manage their workflow. The most straightforward way is the old concept of putting sticky notes on a white board to show the stages of the project development and its progress.

Essentially, Kanban is based on the pull system. Whether you work in a small business or anywhere else on the value chain, pull systems will help you minimize inventories and make necessary changes to the production system as the demand arises. They provide you only with what you need which consequently reduces costs and waste.

Why should you use Kanban?

- Kanban helps you recognize bottlenecks in work processes
- Kanban helps you work on continuous improvement and expand your knowledge through the lessons learned
- Kanban limits multitasking which, in turn, increases the quality of the process
- Kanban can be the fastest way to deliver positive results which will help you gain competitive advantage on the market

- Kanban allows you to re-prioritize your project plans at any given time based on the new information and new business climate or increased customer demand
- Kanban helps your team visualize all the tasks and have a clear overview of all the stages and process which further upscales team collaboration

Is Kanban the right choice for you?

As it was mentioned before, Kanban is an approach which will help you make changes to the management that is designed to meet minimal resistance. Therefore, if you have some processes that are working well, Kanban will help you improve over time without massive and radical change.

Kanban is a great solution for projects of different sizes, complexities and urgencies. Although Kanban works well in most industries, the practice has proven that it works best in product development environment.

Finally, If you want to build trust within your organization and take productivity to a whole new level, opt for Kanban and see how it works for your business.

Scrum

Scrum is a project management framework that helps small, close-knit teams develop complex products incrementally. Scrum focuses on how people work instead of what they do. Scrum relies on agile principles and is the most popular agile methodology out there.

In Scrum, teams develop software in sprints and release what they've worked on every two weeks. This way, customers get bug fixes and new features as soon as they're done, and there's less risk altogether.

What is Scrum

Scrum is the most popular agile project management framework that's used in software development organizations. Scrum can also be used in schools, agencies, government, and other types of organizations.

Scrum was first introduced in early 1990s by Jeff Sutherland and Ken Schwaber.

Scrum got its name from rugby. In rugby, scrum is when a team huddles around the ball and tries to move it down the field in order to win. Scrum is a metaphor meant to reflect how everyone needs to work together to complete the project.

How Scrum works

In Scrum, you deliver to your customer new code and functionality every two weeks. Those two weeks represent one sprint and the whole workflow is built around them. To better understand how Scrum works, we'll talk about what happens before the sprint, during, and after.

BEFORE SPRINT

Before you can start coding, you first need to plan what you'll work on. In Scrum, there are no big master plans like in Waterfall where you lock up your resources months in advance. Instead, Scrum lets you work on one thing for two weeks and then reflect on what you'll work next.

Everything starts with your users/client. First, you get a wish list from your customers in a special format called user story, which looks like this:

As a (role), I want (feature) so that (reason

> *eg. As a manager, I want custom time reporting so that I can calculate my employees' exact salary.*

Then, you create a task for each user story, put it in a backlog, and estimate each task (together with the team).

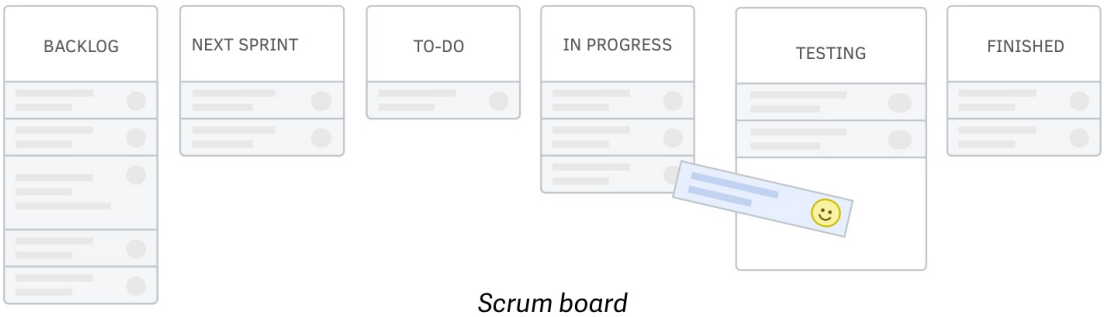
Finally, you decide on what items you will work during the sprint. If some item can't be done in one sprint, it's called an epic and you have three options: split it across several sprints, leave it for later,

or make a new project and form a separate team to work on it.

DURING SPRINT

You set up a Kanban board to visually track progress, see who works on what, and control bottlenecks. The board is usually split into several task lists (columns):

- **Backlog** - all feature requests and bugs go here first
- **Next Sprint** - tasks you'll work after the team finishes the current sprint
- **To-Do** - what the team needs to complete during the current sprint
- **In Progress** - tasks that the team is actively working on right now
- **Testing** - finished tasks that need to be tested before marked as complete
- **Done** - finished tasks that are ready to be shipped once the sprint end



Scrum board

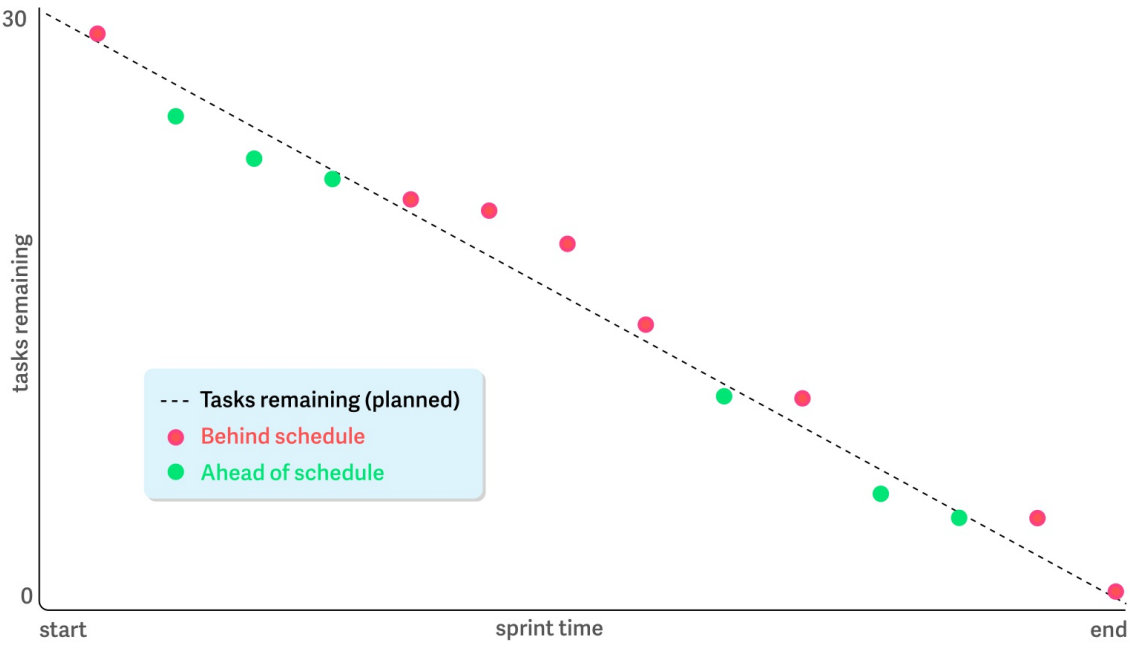
Once you've decided on what items to work, the team pulls tasks

from the To-Do columns and moves them to In Progress as they start working on them. Once they're finished, the task goes to Testing; if the task needs more work, it goes back to In Progress; once the task meets the Definition-Of-Done criteria, it goes to Done and is ready to be shipped.

Each day before work, the team has a daily stand-up). The meeting doesn't take longer than 15 minutes and everyone shares a quick status update:

- Things I have done since yesterday's meeting
- Things I am going to get done today
- Obstacles I need someone to remove

To measure progress, the team uses a Burndown Chart, which is a graphical representation of work left to do versus time. That chart shows how efficient the team is, if they're going to ship on time, and if they need to optimize the process.



Burndown chart in Scrum

AFTER SPRINT

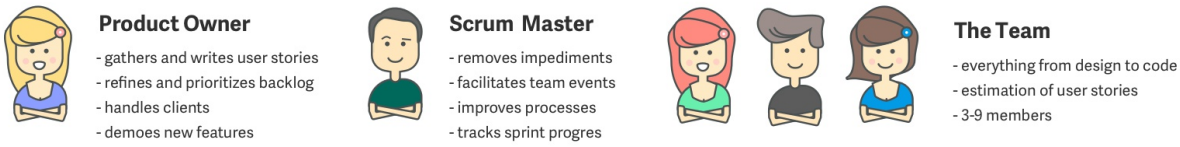
At the end of the sprint, the team reflects on what they've done. They have two reflection sessions:

- Sprint review (2h), where they discuss the work they've done and the planned work they didn't do
- Sprint retrospective (1h), where they talk about the process (what went well and what could be improved in the next sprint)

After the reflection, the team estimates new user stories and decide on what to work during the next sprint.

Roles in Scrum

There are 3 roles in Scrum:



Product owner

Product owner is the visionary and represents the voice of the customer/user. If a team member has any questions regarding functionality of some feature, it's product owner's job to clarify the purpose of that feature.

Product owner focuses on the business side of product

development and spends most of the time talking with stakeholders. Product owner doesn't care about the technical implementation, only the end result.

Product owner's main job is to:

- Gather and write user stories
- Refine and prioritize backlog
- Handle clients and demo new features

Scrum master

Scrum master's role and responsibilities are similar to project manager's, with an emphasis on team facilitation and making sure the Scrum framework is followed.

Scrum master, as opposed to a traditional project manager, doesn't manage people because that goes against agile principles.

Scrum master's main job is to:

- Remove impediments for team members
- Facilitate team events
- Track sprint progress
- Improve processes

The team

The team does all the actual work, from analysis and design to coding and testing. They are self-organizing and work without supervision.

Scrum teams are cross-functional, meaning they don't have clear sub-role and everyone can do anything. Most teams compromise by hiring T-shaped workers so, while everyone can do anything, there is still some specialization and work division.

Scrum work only if the team is small, up to 9 people, max. Any team larger than that changes team dynamics, thus rendering Scrum ineffective.

How to implement Scrum

Scrum is more about about the mindset than some checklist you need to follow to a T. Once you have agile mindset, here's the most common way companies implement Scrum:

1 Pick a product owner

2 Pick up to 9 cross functional team members

3 Pick a scrum master

4 Create and prioritize a product backlog

5 Refine and estimate items in the product backlog

6 Put up a Kanban board

7 Plan the sprint

8 Have a daily stand-up

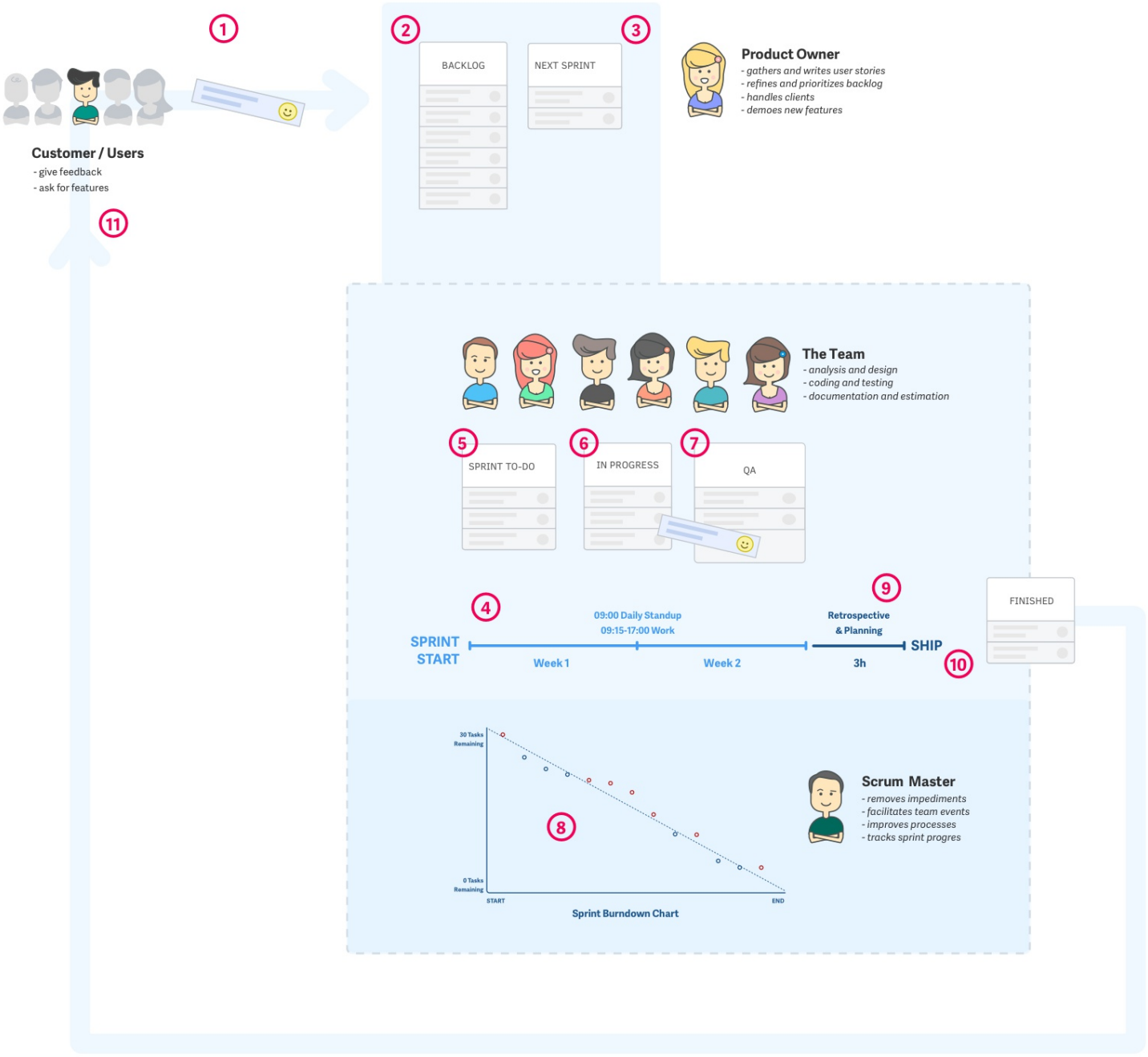
9 Work

10 When finished, do sprint review and sprint retrospective

11 Immediately start the next sprint

How Scrum works

brought to you by active.collab



1. Product owner asks what features they need and why
2. Product owner writes user stories and puts them in BACKLOG
3. Product owner prioritizes tasks and decides work for NEXT SPRINT
4. The team has a 15-minute daily standup meeting
5. Team members pull tasks from TO-DO
6. When someone starts working on a task, they move it to IN PROGRESS
7. Finished task goes to QA for review (then back to IN PROGRESS if it needs more work)
8. Scrum master tracks team progress and efficiency using Burndown Chart
9. When sprint ends, the team discusses the sprint and suggests improvements
10. The team ships what they've worked on to client/customers to use
11. Product owner demoes the finished work and gathers feedback for new user stories and BACKLOG

Extreme Programming (XP)

Extreme Programming (XP) is an agile project management framework used in software development. It prescribes everything, from how to organize projects and develop software, to how to increase developers' productivity and what's the best way to collaborate on code.

Extreme Programming is an agile framework, which means it advocates frequent releases, iterative development, and high level of customer involvement. It's similar to Scrum, only XP is a lot more prescriptive and opinionated.

Unlike Scrum, which is primarily concerned with team organization, XP is primarily concerned with code: its simplicity, how it's written and deployed, how developers work, etc. This means you can apply XP only if you work in a software development company.

What's at the core of Extreme Programming (XP)

The easiest way to understand XP is to think of it as Scrum with an added layer of coding best practices. XP takes agile principles, gives you some processes like Scrum, and tells you exactly how to approach coding.

XP starts with an idea that software development is hard. That's

why, according to XP, the 4 most important things in software development are:

CODING - At the end of the day, if the program doesn't run and make money for the client, you haven't done anything.

TESTING - You have to know when you're done. The tests tell you this. If you're smart you'll write them first so you'll know the instant you're done. Otherwise, you're stuck thinking you maybe might be done, but knowing you're probably not, but you're not sure how close you are.

LISTENING - You have to learn what the problem is in the first place, then you have to learn what numbers to put in the tests. You probably won't know this yourself, so you have to get good at listening to clients - users, managers, and business people.

DESIGNING - You have to take what your program tells you about how it wants to be structured and feed it back into the program. Otherwise, you'll sink under the weight of your own guesses.

> *Listening, testing, coding, designing. That's all there is to software. Anyone who tells you different is selling something. - Kent Beck, the creator of Extreme Programming*

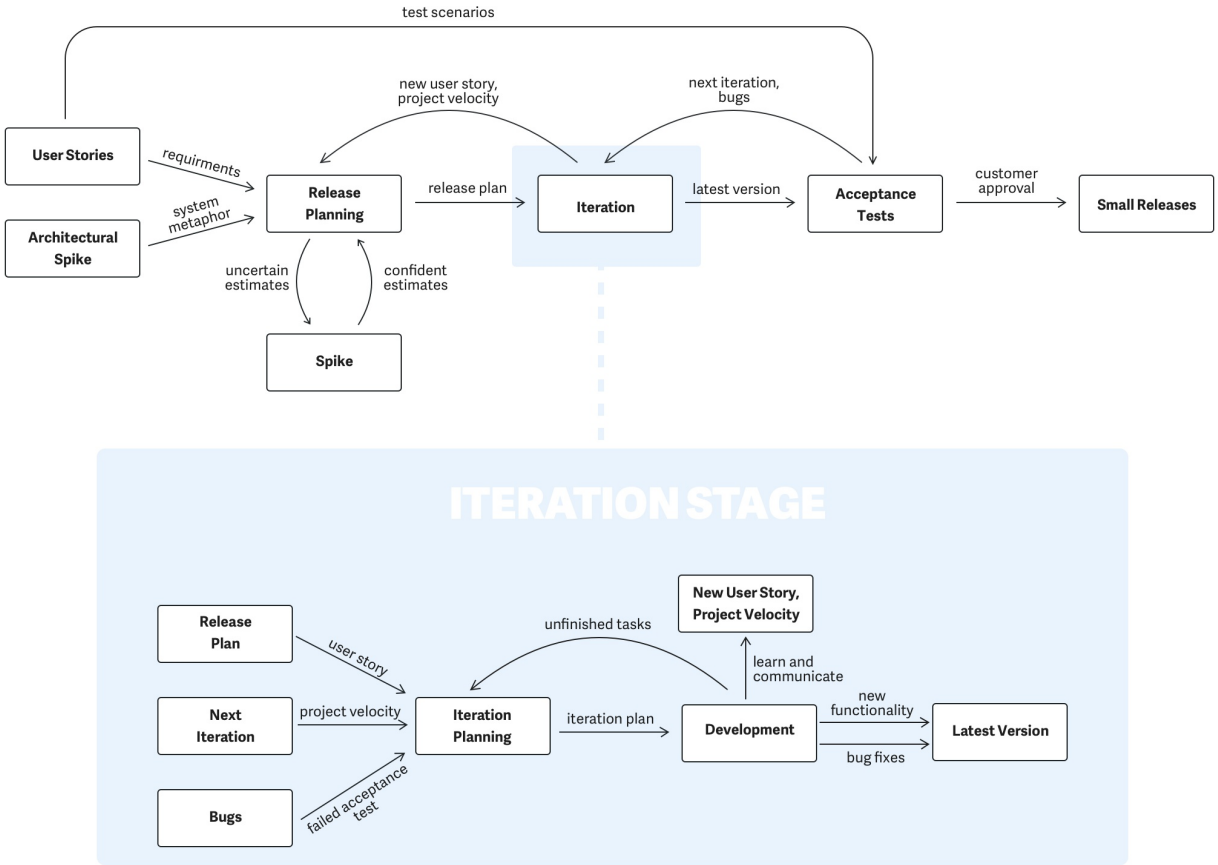
Project lifecycle in XP

Small releases - finished work is delivered to the customer at the end of each week, so the customer gets value sooner rather than

later.

Weekly cycle - the team plans what features to deliver at the end of each week, works in one-week cycles, and reviews progress each Friday. If some feature takes more time, it's broken down into several cycles.

Quarterly cycle - because work is delivered in small releases, you need some long-term strategy to make sure stories fit together and software doesn't end up all over the place.



Extreme Programming project management workflow (from extremeprogramming.org)

Best practices for code collaboration

Pair Programming - all production code should be written by two developers working on a single computer. This should increase code quality and lower number of bugs, thus saving you from future bug fixing.

Collective code ownership - everyone has access to all code and is empowered to fix things on their own. Also, code that you write will be used by someone else so you should make your code clean and simple to understand.

Together but alone - there shouldn't be cubicles and walls between developers to encourage face-to-face communication. But, everyone should have privacy when needed, and shouldn't be interrupted so they can stay focused.

Cross-functional teams - work is done in small teams so each developer should have multiple skills and take care of whatever that needs to be done. There are only two mandatory roles on a team (customer and developer), and two optional (coach and someone who will track results).

Customer always available - only the customer knows how the software should work and only they can answer questions that constantly pop-up during the development.

Information radiators - work should be transparent and everyone should know who is working on what. This is solved by using a

public Kanban board.

Daily stand up meeting - it's more efficient to have one short meeting everyone has to attend than many smaller getting-up-to-speed meetings during the day.

Best practices for coding

The planning game - you should treat planning and estimating like a game, with rules to create emotional distance. This way, both business and development knows what to expect and there's no conflict.

Simple, incremental design - simply designed software takes less time to write and fix than complex one. You should never add extra functionality because only 10% of it will be actually used, thus wasting 90% of your time.

Refactoring - you should always go back and improve code (make it simpler, remove redundancy, eliminate unused functionality, etc). The goal of refactoring is to make old code better in order to make work easier in the future.

Coding standard - all code should be consistent so other people can read it easier. This means, for example, adopting Hungarian notation or camel case, deciding on tabs vs spaces, etc.

Best practices for code deployment

Test-driven development - tests are the cornerstone of XP. Automated testing saves time and lowers number of defects. To make sure developers really focus on tests, they first need to write an automated test (which will first fail), and then develop just enough code to pass the test.

Ten-minute build - you should be able to build the whole system and run all of the tests in 10 minutes. If the process takes longer, it's less likely test will be run on a frequent basis, thus negating benefits of testing.

Continuous integration & deployment - to be able to pull of a 10-minute build, you need to automate testing and code deployment. Every time changes are added to the code base, they are immediately tested and deployed if they pass the test.

When XP doesn't work (and its disadvantages)

XP doesn't work for every software development company. Each company has a different structure and developers a different type of software, so a one-size-fits-all approach XP advocates doesn't work everywhere.

XP suffers from the same disadvantages as all agile methodologies, plus a few more of its own:

- XP is a complex framework with a lot of rules, rules someone has to enforce and everyone has to adopt.
- Collaboration takes more than putting all developers in one

room, and logistically, it's seldom possible.

- XP relies on too many things, so if you don't do one thing right, all others both up and down the dependency chain will get affected.
- XP doesn't offer a clear-cut, foolproof process you can easily adopt and follow
- Having an on-site customer so you don't have to write extensive specs is extremely risky because, if they leave, they take all domain knowledge away with them
- Pair programming locks up twice as much resources as solo programming. Pair programming is useful for solving complex problems, but solving complex problems is very rare in everyday development in 90% of companies. Plus, not everyone like to work in pair and it's more difficult to achieve state of flow.
- XP de-emphasises good UX design. Walking customers through mock-ups, storyboards, and use cases until they know what they want is much cheaper than fixing code. This makes XP especially bad for both large programs and consumer software as good UX design doesn't happen incrementally.
- Refactoring and unit testing is not a substitute for thoughtful design, modeling, and software architecture.
- Constant involvement is a tough sell for your customer/client, as well as absence of knowing how much all is going to cost.

- Refactoring can be a time-waster. Refactoring is only beneficial for code that is used often by others than the original author, which is not always the case.
- Automated testing can only detect "computer" problems, while subtle, "human" usability problems are pushed aside.
- Most software consultancies and contractors don't care for the level of code quality that XP demands. Their job is to simply make something work in the short term, while maintaining it is often someone else's job, thus they have no incentive to refactor or do other unbillable work.

However, the biggest problem with XP is that it's based on the assumption that the whole team is in one place - both development and the customer. In most cases, that's not viable.

Most software development companies today are outsourced, meaning the customer and the development team work not only in different places but different continents altogether, making XP impossible. Just to get an answer about some requirement, XP team would have to wait anywhere from a few hours to a day. This is a deal-breaker for a framework that relies on agility and close collaboration.

Rational Unified Process (RUP)

Rational Unified Process is the ultimate guide how to assign tasks and responsibilities within a development organization and develop a high-quality software that meets the needs and requirements of its users. It was initially created by the Rational Software Corporation which IBM bought out in 2003.

The incredible success of RUP approach has made a wide range of organizations worldwide realize how important a well-defined and well-documented software process is, and stimulated them to start implementing it in their projects.

Although this method is primarily used for big software development projects, many project managers assume that RUP is not the right solution for the projects of limited scope, which don't require many resources. However, there are many examples of small projects that benefited greatly from incorporating RUP into their organization.

For example, TAP University (Technology as Promised) is an online learning management system. Their aim was to extend the face-to-face training and boost their online offerings to corporate and public customers and students.

Although this was a small project, the implementation of RUP yielded remarkable results. It helped the organization build a necessary framework in terms of organization of their use cases

and gave guidance for the company to begin the Construction stage, the third and the most important stage of Rational Unified Process.

What is Rational Unified Process (RUP)?

A processed product

The development team for RUP is working closely with clients and partner groups organizations to ensure that the process is constantly updated

The RUP leverages team productivity

It allows the team to have a free access to acknowledge base with all the guidelines and tool mentors that help them overcome critical issues. This helps the entire team share the same language when developing a software

The RUP creates and maintain models

Instead of producing a large amount of paperwork this method creates models - semantically rich representations of the software system your team is developing

The RUP is a guide how to use Unified Modeling Language (UML)

UML allows your team to communicate their requirements, architecture, and design of the project.

The RUP is a configurable process

It is a simple and clear process that can fit both small development teams as well as large organizations.

Six fundamental practices of RUP

There are six fundamental best practices that make RUP suitable for a wide range of projects worldwide:

- **Develop software iteratively** - by tackling high-risk elements on every stage of the projects, it allows you to gain increasing understanding of the problem and make necessary changes until you reach the most reasonable solution
- **Manage requirements** - describes how to organize and keep track of functionality requirement, documents, trade-off, decisions and business needs
- **Use component-based architecture** - it structures the system architecture into components which are reusable not only on the project at hand but in future projects as well
- **Visually model software** - it shows you how to create a visual model of a software to capture the structure and behavior of architecture and components
- **Verify Software Quality** - it enables you to evaluate and control the quality of all activities during the software development
- **Control changes to software** - it gives you the ability to

control, track and monitor changes which enable constant and successful development of the software. It also helps you create a secure workspace by isolating it from changes which in turn brings your team together making them work as a unit.

The structure of RUP

This approach describes who is doing what, how and when. RUP can be presented by using four main elements:

Workers - the “Who”

It defines the behavior and responsibilities of all team members who are all focus on one common goal - to produce artifacts. In RUP, the worker is more of a role defining how individuals should carry out their work. A worker should not only perform a certain set of activities but also be the owner of a set of artifacts.

Activities - the “How”

It refers to the unit of work that a worker is to perform. Each activity has a clear purpose and is assigned to the specific worker. Activities mainly include creating or updating some artifacts such as a model, a class, or a plan.

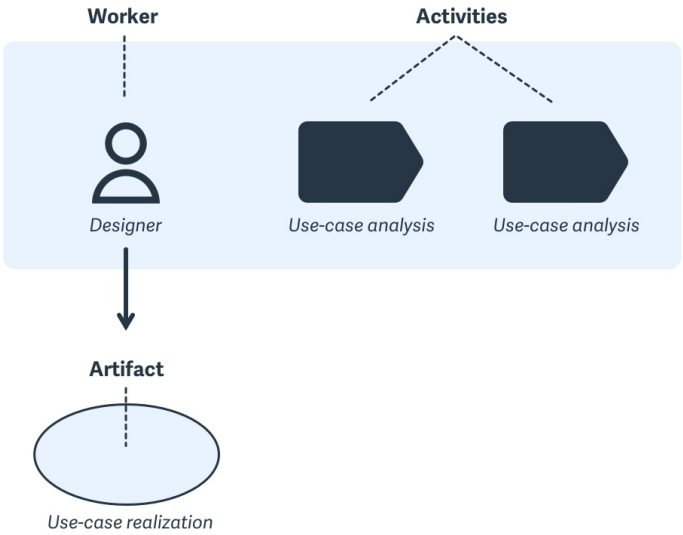
Artifacts - the “What”

The thing or the information that the process produces modifies or uses while working towards the final outcome. Artifacts serve as input that workers use to perform an activity and are also results

or output of those activities.

Workflows - the”When”

Workflow represents a sequence of activities that produce an observable value. In UML terms, we can present workflow in a sequence diagram, a collaboration diagram, and activity diagram.



Workers, activities, and artifacts

The lifecycle of RUP

Like most methodologies, RUP lifecycle is broken into four main phases (cycles), each phase working on a new generation of the product:

1. Inception phase

Essentially, in this cycle, your team determines the structure and the basic idea of the project. Also, the team will decide if the project is worth pursuing at all based on the estimated costs, the

necessary resources and the goal they are trying to achieve with the project.

2. Elaboration phase

The aim of this phase is to analyze the requirements and the architecture of the system, develop the project plan and eliminate the highest risk elements of the project. It's undoubtedly the most critical of all stages as it signifies the transition from low-risk to high-risk. It's an also the point when your team has to make a decision whether to start a construction (development and coding) or not.

3. Construction phase

At this stage, your team is finally ready to develop all components and features and integrate them into the product. It's a manufacturing process where your team focuses on managing resources in order to optimize costs, schedules and the quality.

4. Transition Phase

The transition phase is the moment when the product is finally finished, released and delivered to customers. However, once the product is given to the user, there are a number of issues that can arise. This requires the team to handle all the bug-fixes and correct problems, or to finish some features that were postponed.

At the end of each phase, there is an important Project Milestone - a point in time when your team confirms that certain goals have been achieved. At that moment they have to make some critical

decisions, which will have an impact on the following phase.

The phases and major milestones in the process

The benefits of RUP

- It allows you to deal with changing requirements regardless of whether they are coming from the customer or from the project itself.
- It emphasizes the need for accurate documentation.
- It forces integration to happen throughout the software development, more specifically in the construction phase.

The disadvantages of RUP

- It mostly relies on the ability of experts and professionals to assign the activities to individuals who should then produce pre-planned results in the form of artifacts.
- The integration in development process can also have an adverse impact on some more fundamental activities during the stages of testing
- Although RUP has delivered excellent results, especially in software development, it is a rather complex method which makes its implementation challenging, particularly for smaller businesses, teams or projects.

Crystal Methods

When in 1991, IBM asked Alistair Cockburn to develop the methodology for object-oriented projects, he knew it will be a real challenge as he didn't know much about project methodologies at the time. So, he decided to interview project teams and find out their view of the project.

After doing an extensive research he came to the conclusion that tall successful teams shared the same patterns and techniques without even using any specific project methodology. In other words, they added value to the aspects like close communication, morale, access to users and others, which you can not find in any specific methodology.

Finally, he used his findings to construct a family of methodologies and named it Crystal.

What is Crystal Method?

Crystal method is an agile software development approach that focuses primarily on people and their interactions when working on a project rather than on processes and tools. Alistair believed that the people's skills and talents as well as the way they communicate has the biggest impact on the outcome of the project.

Crystal Method is based on two fundamental assumptions:

- Teams can streamline their processes as their work and become a more optimized team
- Projects are unique and dynamic and require specific methods

According to Cockburn, we should view the product development as a game which should stimulate everyone to interact, become creative and produce brilliant ideas. He says that instead of focusing on questions like “is our model accurate?” we should be looking for answers to the questions like “Is our product meeting the customer’s needs? Or “Do we have our goals aligned as a team?”

Crystal Method family members

One of the things that Cockburn discovered is that the project properties changed depending on the number of the people involved in the project and the level of criticality of the project at hand.

While the smaller team can handle and build the product without a lot of status reporting and paperwork, the number of “communication artifacts” rises with bigger teams who are working on large-scale projects.

In other words, the more people you have on the team, the more critical the project is and the more complex the approach needs to

be. Therefore, there is no one single Crystal method; there are different Crystal methodologies for different types of projects.

To make this categorization easy to understand, Cockburn named the methodology Crystal and categorized it along two dimensions size and criticality that matching those of minerals - color and hardness.

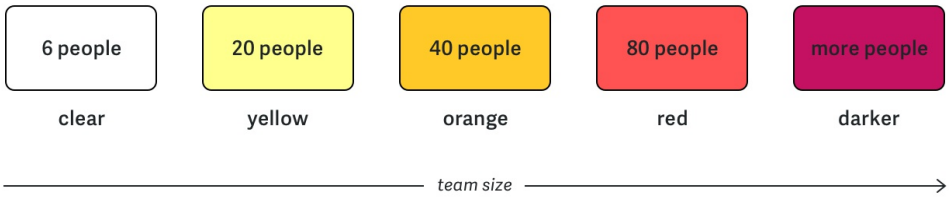
Essentially, Cockburn developed these families to point out that each project may require a particular set of policies, practices, and processes in order to meet project's unique characteristics. Cockburn tried to explain this by calling Crystal "a set of samples that you adjust to your circumstances".

Which approach will be most suitable for your projects depends on three dimensions:

- Team size
- Criticality
- What the priority of the project is

Generally, they are characterized by colors, according to the number of people involved in the project:

- **Clear** - for teams of 8 or fewer people
- **Yellow** - for teams of 10-20 people
- **Orange** - for teams of 20-50 people
- **Red** - for teams of 50-100 people



Crystam Methods colors

Crystal Method characteristics

Human-powered

This means that people involved in the project are vital and that the processes should be adapted to meet people's needs. It also emphasizes that people are capable of organizing themselves and that they can become more organized and competent as the processes develop.

Adaptive

Crystal is a stretch to fit methodology meaning that processes and tools are not fixed, but have to be adjusted to meet the requirements of the team and the project at hand.

Ultra-light

Crystal doesn't involve too much documentation, management and reporting. It keeps things light by focusing on transparent workflow between the team and the client and by practicing open communication between team members.

7 properties of Crystal Method

- **Frequent delivery** - it allows you to frequently deliver

working, tested code to real users. In this way, you don't have to face the fact that you have invested your energy and time into the product that nobody wants to buy.

- **Reflective improvement** - no matter how bad or good the product is, there are always areas where the product can be improved. Also, there are always new techniques and methods your team can implement to improve their future practices

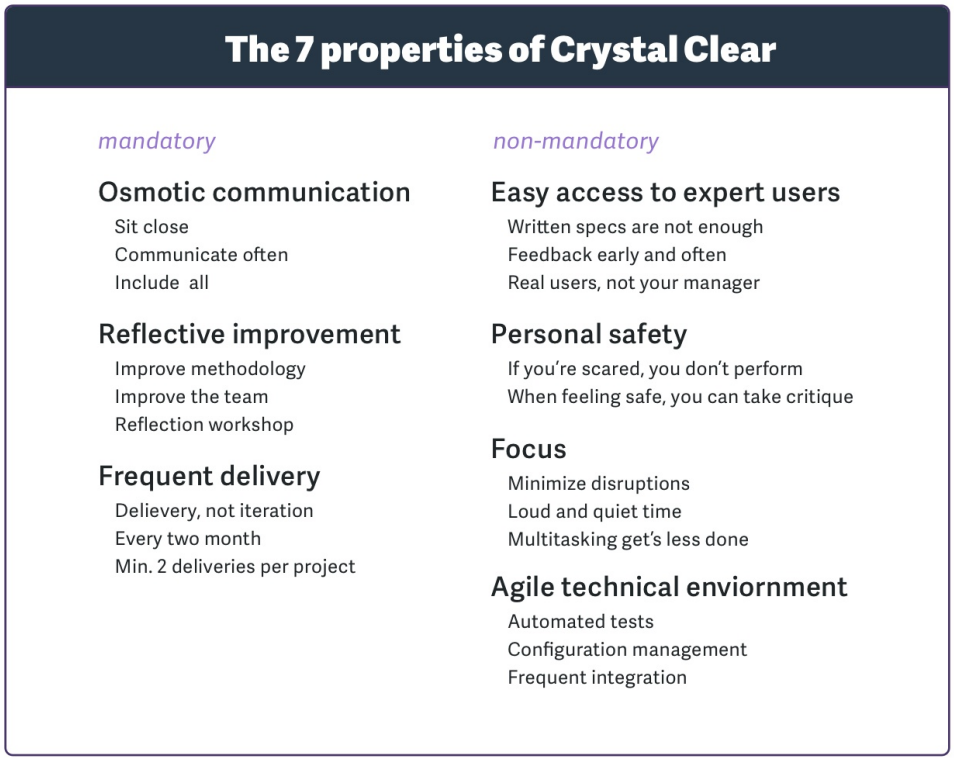
- **Osmotic communication** - with the team who works co-located, information flows around the team. That allows them to pick up valuable information without even being directly involved in the discussion of the certain matter.This gradual absorption of ideas is called osmotic communication. Cockburn believes that this kind of work atmosphere can operate with very little structure.

- **Personal Safety** - the only way to build a healthy working atmosphere and a true team culture is by practicing an open and honest communication. Team members should be able to speak without the fear, no matter whether they are presenting a new idea or talking about a potential problem.

- **Focus** - each team member knows exactly what to work on which enables them to focus their attention and avoid switching from one task to another. Also, this boosts team communication and helps the team prioritize and work towards the same goals.

- **Easy access to expert users** - Crystal enables your team to maintain communication and get regular feedback from real users.

- **A technical environment with automated tests, configuration management, and frequent integration** - very specific tools for software teams where the emphasis is on continuous integration so that the errors could be caught within minutes.



Why is Crystal Method useful

The fact that Crystal uses a focus on people and communication as its organizing principle I what distinguishes it from other software development methods.

Unlike other agile methodologies, Crystal focuses on adjusting the techniques used in a project with the aim to strengthen the

process of team communication. Besides that, Crystal allows:

- Continuous integration
- Flexible and configurable processes
- Active user involvement

Keep in mind that Crystal is primarily created to remind you how to stay centered and focused on your work during the project development.

This means that you need to have an extensive knowledge and wide experience in software development if you want to successfully implement it and triumph in your next future projects.

Feature Driven Development (FDD)

Feature driven development is a process that provides businesses with feature-rich systems that should help them control their ever-evolving nature.

As the name suggests, features are an important aspect of the entire Feature Driven Development (FDD) process. FDD blends a number of best industry-recognized practices which contribute to the business by complementing and reinforcing each other. While these practices are not new, their particular blend is new.

The chosen practices are all client-based. This means that the developers focus on the features that their clients value and find important for their business. On top of that, these features help the developers handle each step of the planning stage of the project development.

The birth of Feature Driven Development

Back in 1997, when Jeff De Luca was the Project Manager of a large software development project for a bank in Singapore, he encountered the complex problem. He realized that even by using all the available resources, his knowledge and traditional strategy of software development he could not solve the problem.

He consulted Peter Coad and they came up with the concept of

feature driven development together. The concept consisted of five processes that were designed to cover the model's development, its listing, design, planning and the feature building. In print, this was first published in the book called "Java Modeling in Color with UML" written by Peter Coad (Peter, et al., 1999).

Although FDD was Initially aimed at for smaller teams, De Luca and Coad designed the feature driven development from the ground-up to work for larger teams. This was of huge benefit to not only small businesses which are just learning the ropes of project management, but also huge corporations who often have to fight challenges of working with large teams and organizations.

The 5 processes of FDD

Jeff Luca proposed a solution: a mix of five processes that would cover the development of the model.

Developing an overall model

Domain and the development team members work together to create an object model of the domain problem. Their goal is to propose the model for the domain area. They are always under the watchful eye of a Chief Architect who also gives them guidance. After the teams have proposed their own models, one of the proposed models or a merge of models is selected and it becomes the model for that domain area. This helps the team have a clear overview of the entire project.

Building a feature list

Once your team has developed an object model, the next step is to identify the features that are valuable to the client. The features are the building blocks of the project and help the team navigate the processes.

Planning by feature

The third stage revolves around organizing the features and the way your team will implement them. Naturally, it's important to take team workload, risks, and other aspects into consideration so you can prevent any complex issues from arising.

Designing by feature

By using the knowledge gained in the first modeling process, the chief programmer selects the group of features the team should develop next and determines the domain classes. Once the team starts working on the project, the domain expert starts analyzing and designing a solution to each feature.

Building by feature

The next step is to implement all the necessary items to be able to support the design. In other words, after your team has developed, tested and inspected the code, they are ready to start building the software.

Why do we need FDD

As the software system grows, the software development becomes more complex. In such a scenario, your team will have to find the most efficient way to tackle the pains they come across. FDD is the method that allows your team to successfully runs the project by:

- **Improving the communication** - most processes require constant communication flow, especially in large teams. FDD allows team members to communicate more easily while encouraging team creativity and innovation
- **Minimizing the complexity of the system** - the bigger the size of the software system, the more complex the system gets. Complexity is one of the most frequent obstacles every software developer needs to overcome because it quickly outstrips the capacity of the human brain. FDD gives your team the ability to break the entire problem into many smaller problems which they can deal with in a smaller period of time. On top of that, smaller problems minimize the communication needs within the team and improve the communication flow.
- **Maximizing the quality** - each member of the development team perceives quality in a different way. An average user associates quality with the interface, reliability and the response time. On the other hand, a developer talking about the quality discusses the ease of maintenance and enhancement of the software. However, the ever-increasing demands of the market prevent us from predicting the challenges we will have to struggle with and the changes we will have to make to produce positive results.

That’s why when working on a software development, it’s always best to view quality as a spectrum rather than a separate unchangeable parameter. In FDD, the concept of quality does not only include the testing of the code - it also includes coding standards, measuring audits and metrics in the code.

Feature Driven Development vs Scrum

When comparing Scrum methodology to FDD, it is obvious that they have many common points. Both of them:

- Improve communication
- Emphasize quality components
- Enhance collaboration

On the other hand, unlike FDD which focuses on specific engineering practices, Scrum does not specify any particular engineering practice. Also, while FDD has longer feedback loops Scrum has short feedback loops and focuses on a vertical slice.

Benefits of FDD

There are a number of reasons why Feature Driven DEvelopment is gaining popularity in IT world:

- FDD is an excellent solution for big and complex projects especially when dealing with critical situations
- 5 processes help the new team members become familiar

with the system in short period of time

- Frequent progress reporting that takes place at almost any level of the project development, allows your team to keep track of the progress and track results
- It allows you to keep your project up-to-date regularly, identify any errors, and provide your client with valuable information at any time
- The deeper understanding of the software system allows you to build small parts one by one. This reduces the risk and keeps you safe from any unpleasant surprises.

Joint Application Development (JAD)

Joint Development Application (JAD) is a very common technique in business analysis world. It brings system developers and users together in a productive and creative environment through a structured approach that involves discussion groups with the goal to obtain requirements and specifications.

In a nutshell, the ultimate purpose of JAD is to include the client in the development process and develop a more satisfactory end-product that will meet the client's needs more efficiently. JAD allows clients to have full autonomy in project development and allows them to participate in their application's development through a series of workshops.

When business needs some technical input from the technical expertise, JAD session is required. Likewise, when technical expertise needs business input, JAD is required.

Whenever a business and a technical team are trying to collaborate with each other to make certain decisions about critical issues, there is nothing more effective than organizing a JAD session.

Evolution of Joint Development Application (JAD)

Chuck Morris of IBM Raleigh and Tony Crawford of IBM Toronto

developed Joint Development Application in the late 1970's with the aim to improve client satisfaction.

During the 1980’s they led few workshops to prove the concept. Since then, it has been applied to numerous software project across industry sectors and has produced excellent results, particularly on small to medium-sized projects.

JAD participants

When implementing JAD process, you need to keep in mind that the JAD team is at the heart of the project. Also, you need to carefully select and include your stakeholders as they will be responsible for the overall success of a JAD session.

Here are key roles you need to involve to keep your JAD session on the right track:

Executive sponsor

This person is usually the manager of the business area who comes from the customer's company and has the full freedom to make critical decisions concerning the project.

Although they don’t have to actively participate in all steps of the project, they need to be available throughout the process and solve important issues as they arise.

Facilitator

This is the most important person in the process as they are responsible for planning, executing, and managing the session. The facilitator should have the right knowledge and extensive experience to lead the project.

Also, they should work closely with executive sponsors to achieve desired goals. During the discussion, the facilitator should be able to:

- Focus on the process
- Be unbiased and neutral
- Lead groups and keep sessions on track
- Stop sideline conversations

Stakeholder

Stakeholder is the main focus of the entire process. Without their involvement, JAD sessions are pointless.

They represent all key user groups affected by the project development and represent multiple levels within the organization. JAD session allows stakeholders to become an integral part of the project so they can get the product they need.

Scribe (Recorder

Scribe is in charge of documenting the entire JAD process. Since there are often a lot of ideas and suggestions, a JAD session may involve more scribes. A Scribe must:

- Capture the important decisions, who made them, and why
- Distribute and archive the documentation at the end of each session
- Have excellent analytical skills to be able to analyze the discussion

IT Representative

IT Representative gives technical advice and helps JAD team develop logical models to build a prototype. They must:

- Help the customer turn their concepts into business requirements
- Efficiently use available technology
- Provide end solutions that are realistic for the budget and timeframe

Observer

This person observes each step of a JAD session, end-user's needs, and JAD sessions decisions

JAD sessions

Each JAD session must have well-defined objectives and detailed agenda items. You need to ensure you have invited key players from both the business and technical world, as well as the person who will take notes.

You need to ask questions, record important decisions, and assign action items. While the questions and agenda items are the essence of the discussion that will drive the meeting, don't expect to get the answers immediately.

The point of a JAD sessions is to trigger creative thinking and brainstorming that will lead to a joint discussion that will require expertise from various departments.

At its core, JAD session should help the teams come up with decisions. If you and your team can't arrive to a decision, you need to run JAD sessions on a scheduled basis, which are usually known as JAD workshops.

The outcome of a JAD session is more valuable than other meetings because because you can make more decisions.

Although most of the JAD sessions are scheduled during the development phase, they may even happen during the requirement phase.

Advantages and disadvantages of JAD

Advantages

- JAD allows you to resolve difficulties more simply and produce better, error-free software
- The joint collaboration between the company and the clients lowers all risks

- JAD reduces costs and time needed for project development
- Well-defined requirements improve system quality
- Due to the close communication, progress is faster
- JAD encourages the team to push each other to work faster and deliver on time

Disadvantages

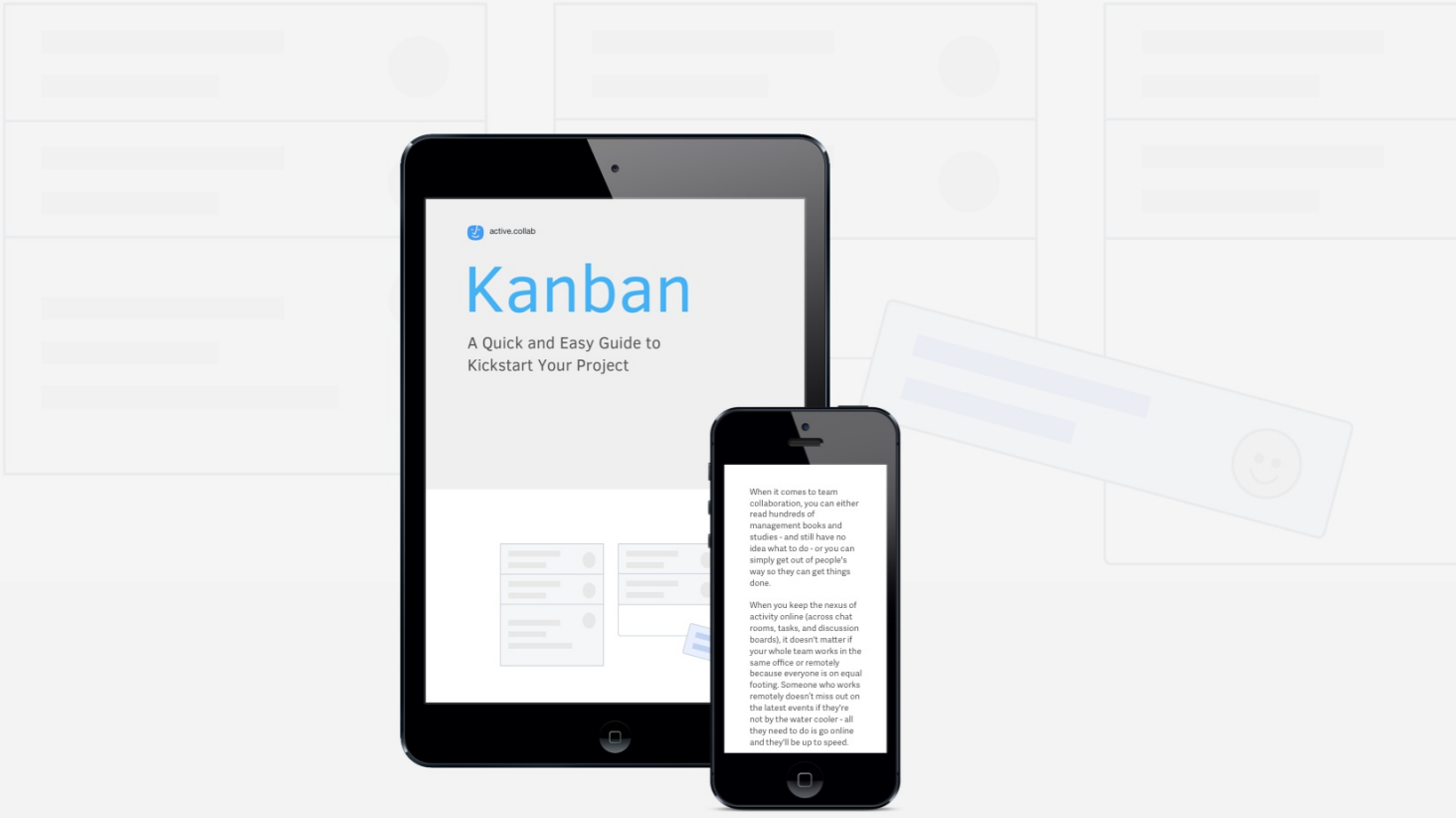
- Different opinions within the team make it difficult to align goals and maintain focus
- Depending on the size of the project, JAD may require a significant time commitment

Further Reading

Kanban: A Quick and Easy Guide to Kickstart Your Project

This book introduces Kanban and key principles of agile project management designed to improve your productivity. The book is very short and is geared towards beginners.

The book will help you learn how to organize projects and how to introduce a simple and reliable process so you don't have to worry about things breaking apart when you grow.



Download

The Complete Guide to Managing Digital Projects

This book dives deep into project management. It covers everything from client collaboration and project management to invoicing and time tracking.

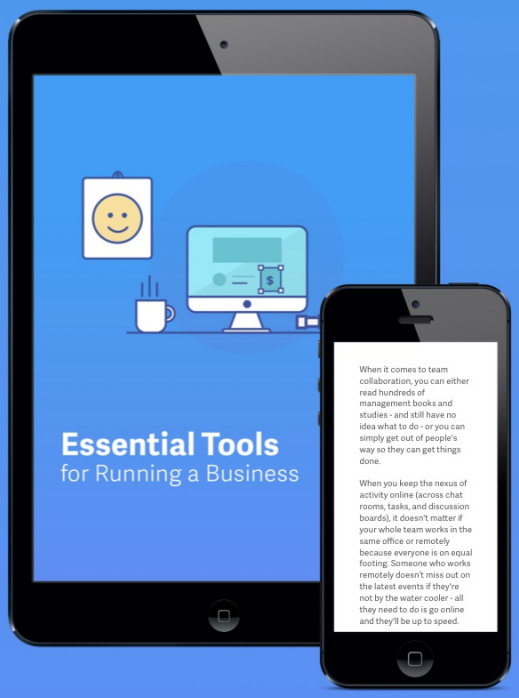
The book will teach you everything you need to know to successfully manage digital projects, get paid, and make your clients happy. Unlike the Kanban ebook, this will take you much longer to read but it's still very easy to understand.

Download

Essential Tools for Running a Business

Every growing business needs tools. This book lists every tools that helped us grow our company from 3 to 30 people (and beyond).

We share behind-the-scenes insight, how we use every app, and how each app can help YOU become more productive. Every tool is illustrated with screenshots so you can see how it works.



Download

GROWTH: Everything You Need to Know Before You Can Grow Your Business

Growth causes processes to burst at the seams. What used to work fine now causes bottlenecks because more and more decisions have to go through the business owner.

If you don't change how you work, that growth is unsustainable and will only make thigns worse. The book covers everything you need to know to avoid mistakes business owners commonly make when growing their business.

Download

The Big Book of Team Culture

Learn about what makes a great team, how to improve teamwork, what it means to be a leader in a modern workplace, and how to create positive team culture - all in one one place.

Download

active.collab

Active Collab is a powerful, yet simple project management software. It helps your team stay organized when you outgrow emai. It's a one-stop solution for all your business needs.

Active Collab is perfect for growing teams. It gives you an overview of your team's activity across projects. WIth it, you can delegate tasks to your team, keep information in one place, estimate and track time, and issue invoices.

For more than 10 years, over 200.000 people have used Active Collab, ranging from small businesses to Fortune 500 members, universities and government institutions.

activecollab.com