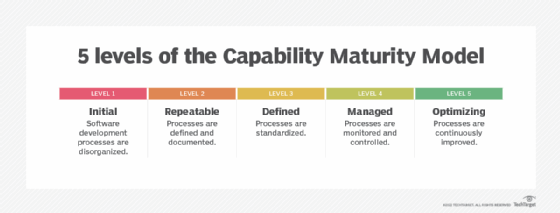
The technologies used have advanced so much that there are multiple inbuilt milestones in the technology themselves. This restricts your work and does not allow you to venture on your own. Because of this, the notion of Reuse is getting ever more popular since the issue of planning forward is getting smaller and smaller.

Stakeholder engagement, risks, and planning are the most important aspects that a Project manager must address immediately. Planning and risk management go hand in hand since one needs to plan and calculate the future risks which will both benefit the project and the team in the future.

View this [model](https://l.messenger.com/l.php?u=https%3A%2F%2Fwellingtone.co.uk%2Fwp-content%2Fuploads%2F2021%2F03%2FThe-State-of-PM-2021.pdf&h=AT17ulXQmFL5S-gemF_BFMvjzaHCb19xUTVBHnr-mi5M47nKeMsijeJjhZ9Flm5GANrMo97poGsVQmowK-q4RGGXqcUtk8FYimHFXbpYFfxwI9CVArmMAQbqGyr8aeHJVgtpxg) for graphs.

PPM Maturity – Measure of maturity of the project. This is based on CMM (**Capability Maturity Model**). This model is based on 5 levels of maturity;



In level 3, **Defined**, the above model is the first time when we can say that most of the processes are understood and the overall project is *mature*. The specifications; people, input, and outputs are well-known and understood.

**Managing** goes a step further, by stating that not only are the processes known but how they interact with one another is also understood. The process is understood enough that they can be *combined* and not have only pieces of the overall project but bigger and more important aspects.

The final level, **Optimizing**, is a stage where the processes are understood so well and can integrate that we can *improve* upon it.

When we have unsatisfied PM maturity, it can be a result of two things;

* The manager is not able to handle and use the tools given to establish the correct maturity.
* Time problems.

***The major issue lies in the fact that people have a hard time understanding how the many pieces of a project come together to achieve a satisfactory and mature project.***

People do not always have the correct same definition of **success**. The criteria of what a successful project is must be specified beforehand since it is different for all companies.

Best definition – Using the different resources provided, human and inhuman, to provide the high-quality software possible (Quality is always crucial in project management). Even with the best resources possible, a pad project manager can still fail and break a whole project.

**Not all managers are managers** (?). This is in the scope that there are different levels of management, in the fact that some managers manage high-level commercial software while others are much less proficient and complex. These managers are not the same and although they may be both successful in their field, does not mean that they are compatible with your company or the work you are trying to do.

If you don’t carry out management in an educated/scientific way you can end up doing things wrongly. One must keep in mind that there are multiple ways to use the resources given and it is a very crucial skill to plan how they will be used for the best quality.

People are the most important of these resources since not all people will behave the same when assigned their roles. The manager must keep in mind that the only thing that matters is the final project. The people have jobs and roles to fulfill but the ultimate goal is the quality of the project.

**What needs to be managed?**

It all depends on the project, and on what you are trying to build. What you are managing will also influence this. There are multiple different groups of people who take part in any project and the manager must be able to handle all of them. Of course, the most important thing to manage is the project.

**What is a project?**

A project is considered in terms of novelty, size, planning, and others. It all depends on what the project is and whom the people are involved with if it can be considered a project or not. Projects are composed of tasks. (View slides)

In general, a project will contain multiple systems which in turn are made of more sub-systems. At the bottom levels, there are only modules. The lower the size, the lower the overall sophistication.

In a software project, progress is not always clearly visible since there are multiple steps (models, cycles …). This is especially the case if multiple people are doing different jobs. This makes it hard to make overall progress.

Levels of abstraction are intelligent tools that humans use to scale down the given product to the end user. It allows the users to comprehend only what they need and not what makes the functions they need possible.

Due to the multiple moving parts that constitute a software product, the cost is sometimes more complex to estimate than in traditional projects. However, the software is unpredictable and it is indeed possible to get the cost wrong (over/underestimation). When unexpected things happen, unexpected outcomes are a given since you can’t predict them. In software, this is very popular since the software is always changing and evolving, with people getting on different trends every day. In reality, the software is most of the time overpriced because of this factor.

Don’t expect to build something that can’t be deployed in the real world.

**What are the parts of the project that we manage?**

* Feasibility Study (Is it worth doing?).
* Plan (How do we do it?).
* Project execution (Do it!).

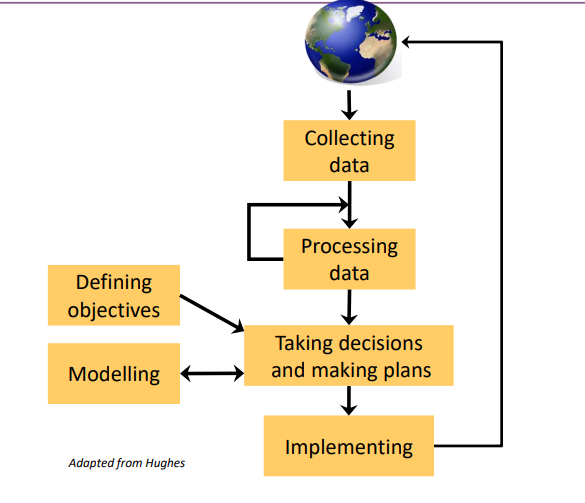
The more the manager interferes in the end, during the execution, the more we can determine that the project has not been planned out correctly. This shows that the manager has not done the required work beforehand. In the end, the manager’s only job is to make sure that the plans and guidelines established are being followed.

* **Planning**: Assigning the given resources to their correct usage to achieve the highest quality software possible. This is done in consultation with the rest of the people involved in the project. A manager that does not consult will risk the entire project.
* **Scheduling**: How to get the tasks to work together so you don’t end up with people waiting on certain tasks to be finished to proceed. The right people must be allocated to the right teams and given the right jobs to finish and work for the union.
* **Monitoring**: Visibility of the process is of crucial importance. This is required to see what has been done and what must still be done.
* **Directing**: This should be done from time to time. This is done to remind people of the direction that **we** (everyone and not only the PM) have agreed upon. Due to the nature of the developers(they know what they are doing and have the experience), the PM can’t keep changing the directions and implementing new notions because they can easily challenge your claim.
* **Controlling**: This is much like the previous, but more severe since this needs interfering. This should be minimal, for example when things are really bad and are threatening the whole project.
* **Organising**: Making sure that the resources present are available when needed.
* **Staffing**: This is one of the most crucial aspects and it needs to be done on multiple occasions. This includes drafting the appropriate teams, which are made up of the correct people. The most important difference between humans and technology is the presence of intelligence. Humans are the most important asset.
* **Innovating**: Being able to look at the real-world process and being able to produce digital solutions to address the given needs. Managers need to be able to adapt in innovative ways.
* **Representing:** This can be divided into two PR (Public Relations) and CR (Client Relations). PR is bringing in new customers, while CR is about retaining new customers. This highly affects how you deal with both groups.

***Project lifecycle.***

Projects are organic, it is something living. Since people make up the project and they are very subjective (which is worse when subjective people are working with each other), it seems like a living thing. The project is the team. How the project/team evolves behaves like the project itself is a living thing.

The right data sets are required. The correct data will give the project meaning and will shape it into what it is. The project should be built around the data sets, data always comes first. When the correct data is obtained, it can be processed into meaningful value. Without the right collection of data and processing, the project is doomed to fail.



When taking decisions and making plans, we start to see how are we going to build the overall project. When you have a defining objective, they are the constraints in the real world. Sometimes these constraints are known as non-functional requirements. Modeling is used to represent what we want to show. It is two-sided since it can be used to understand what you are trying to build, and also to explain what you are building.

In the end, the implementation should be the place where the PM is least involved. If the other stages were done correctly with the required planning and scheduling, the PM only needs to monitor the process. After the implementation, the project will be present in the real world where it will **require** maintenance.

**Project Planning**

The **scope** of the project is the main goal and the final deliverable. This does seem obvious but since there are a lot of moving parts in any given project, the scope might be lost in the process. The main method of doing this is by constantly referring to the **requirements** and making sure that each time a new function is being implemented, a requirement is being fulfilled. If throughout the process, the team realizes that some requirements can’t be met, meetings with the client must be held to reach an **agreement**. The scope is based on *requirements and agreement*.

Project **estimates** are harder than one realizes. A lot of time that a project manager uses, is estimation, trying to plan and *predict the future* which is very hard. This is what makes it such an important tool. The main problem with estimation is that it is solely based on the accuracy of the **parameters** that you set to get your estimation. If these parameters lack **accuracy**, the whole estimate will be false. The more **experience** the PM has in the area/field of the project, the more accurate these parameters might be and the better the development of the project will go. The main things that we estimate are:

* Effort.
* **Cost** (Most important of the 4 since the prediction will mainly depend on cost).
* Time
* Human Resources (The other resources (hardware/software) must also be taken into consideration).

**Risk analysis** is also a very important aspect of project planning. One way of looking at risk is the *probability of occurrence*. **Severity** is another way of looking at risk. This refers to how severe the risk is if it comes to be.

A critical path through a project refers to the tasks that are the most pressing in terms of *schedules and deadlines*. A manager needs to identify the tasks that are on the critical side of the project to **prioritize** them.

Staff organization is also a very important aspect of project planning. You can have the highest quality of resources, *if the people present do not have the right skills and have low morale, the project is doomed to fail*.

When the levels of **abstraction** are affected, all the above aspects are also affected. For example, risk management will come more accurate since things have become more **tangible** and can be understood better. Another thing that can be defined as the development moves forward is the **estimates**. This is because the project will be *more mature and clear*. This will help the PM to create more **accurate parameters**.

When comparing the estimates that were done throughout the project, a later estimation should be **within a previous one** since, if the planning was done correctly, it should have been already estimated. If the PM does do the allocated work before the project even begins, or if they are not inclined enough in the project’s field, the primary estimations will be **inaccurate**.

Other plans

Configuration management is very important since it plans for the **release** of the finished product.

Staff development is the managing of the staff’s morale. If the team is not happy, some might leave which will leave the **project unmanned**. The team needs time to build the correct relationship with one another to start working as a **unit** and not as **individual members**. If team members are constantly changing, the team would need extra time in each instance to become **comfortable** with each other once again.

***Exceptions*** are the things that a project manager is not able to handle/estimate since it is out of his/her hand. The more **robust** your processes are, the harder it is to affect them since they can **adapt** and **counter**.

The project does not end when you deliver the final product, since **maintenance** is a ‘**when’** and not an ‘**if’**. It is important to keep in mind that projects must be built with maintenance in mind since it must and will happen.

Following the above-mentioned procedures does not **guarantee** success. It just helps the project manager to get a hold and control of the situation.

If the team is not motivated enough and just thinks of themselves as another resource, the right mindset will not be achieved. The goals of each member must be the same goal of the project and the motivation of the team to achieve this must be adequate. It is important however to make sure that the goals are accomplishable, and not self-acquainting.

It is important to understand the risk according to what you are building. This will allow us to judge which risks are most likely to happen and how probable they take place. This will allow us to plan what the action and solution are if one of these risks arises.

Risk also plays on the balance of return on investment. This is mainly because the contingency plans that need to take place to counteract what **might** happen will cost you much of your resources.

**Managing people**

Peoples are the most valuable resource that any Project manager has in their toolset.

Given that the psyche drives the mind and overall behavior of the person, if not treated right, the quality produced will not be as good as it could be.

Like everything, the tasks provided need to be broken down to be understood better and understand what kind of resources need to be allocated to the given task.

The best way to make an employee’s life hard is to assign tasks that you know he does not have the skills to accomplish.

When people are motivated enough, even though they are spending more money to accomplish this, both the quality of the work produced and the overall state of the person would be elevated. This is a win-win situation.

There are other ways to motivate employees. These include flexibility, aid when needed, and a welcoming and safe workplace.

*Project Management – Ivan Bartolo Lecture*

**People + Process = Outcome.**

Project management is about managing the budget, tasks, time, risks, benefits, deliverables, etc. It is no use to try and manage people. It is important to talk with the people which work with you and build a solid relationship with them. This is the closest thing to managing people. You also need to learn how to empower and trust people.

When starting a project, the end must be in mind to have a clear goal along the process. Never base decisions on the current position/situation you are in.

When companies invest in software, they are not buying the software itself, they are buying the **benefit** that it brings with it.

The project manager is responsible for delivering a solution, not a system. The people who commissioned him are after a solution to their current problem, and that is what needs to be delivered.

When you are creating objectives, make sure that they are smart and obtainable objectives. The acronym SMART can be used to refer to this: **S**pecific, **M**easurable, **A**chievable, **R**ealistic, and **T**ime-bound.

Project managers need to deliver a solution, not just the software.