1 Summary

1.1 Fundamental concepts

The main idea behind the Capability Maturity Model is that historically it has been very difficult for develop software in a way that consistently meets the companies expectations going into the project. Through paying attention to the process being used by an organizations development teams and continually improving the process teams can more accurately predict the outcomes of projects, and more consistently improve performance on future projects, thus making their performance meet their capability.

Organizations at low levels of process maturity are much more dependent of the skill and motivation of key employees and are less able to hire and train new staff, while mature organizations have a better ability to hire, train, measure, and improve employees making them less dependent on any one person and more resilient to staffing changes.

1.2 Levels of Maturity

The first level of maturity is the initial level. At this stage there is no set process and development can be very unpredictable. Success of a project is very heavily dependent on the skill of the people working on the project. There is no method for code review or verification so teams often find themselves running around trying to fix problems rather than sticking to a set plan or schedule. This results in lots of projects taking too long and going over budget.

The second level of maturity is the Repeatable. Here there is a institutionalized management process for projects. The managers are assigned to projects based on experience with similar projects in the past, and project outcomes are starting to become more predictable. Standards in projects are now being set and enforced, planning and tracking are now a stable part of the process allowing the organization to learn and improve from shortcomings.

The third level of maturity is the Defined phase. At this level and standard process for both development and management is defined, documented and implemented across the entire organization. To go with the organization wide policy a new training program is implemented to train developers and managers about what is expected of them in their role in the process. Everything has to be well defined in order for the team members to meet the process standards. A method of work verification is implemented, like peer-reviewing code, as well as criteria for deeming a project complete. In this stage it becomes easy for people to gauge the progress and status of any project, and outcomes become much more regular.

The fourth level of maturity is the Managed level. This phase focuses on putting mechanisms in place to record data about your development performance and making these mechanisms more consistent. This allows you to better track the actually performance and try to keep outcomes within a acceptable variance range.

The fifth level of maturity is the Optimization level. This stage focuses on using the data obtained from level four to measure how alterations to your process affect outcomes. Regular cost benefit analysis is done to sort out changes that will actually improve the process capability, from statistical noise. In this stage the already clearly defined process is constantly being micro adjusted and improved.

1.3 Improving Maturity

The goal of advancing the maturity level of an organization is to improve the infrastructure around the development teams so that and managers can better communicate, problem solve, and measure the projects throughout there lifespan. Through increasing the maturity level more tools and practices are put in place to make this easier and more consistent, allowing for more analysis and improvement. Each level of maturity puts in place the tools that will be needed to reach the next level, so advancing the maturity makes further improvement possible.

2 Class Discussion

2.1 Why Have Multiple Levels

Our group agreed that the purpose of the different levels was to represent the real world range of maturity that exist, and to provide a graduated selection of attributes the make the different levels of maturity. This allows for organizations to review their process and compare it to the list to find where they are. It also then allows the organization to more easily identify the easiest steps to make in order to meaningfully advance their process maturity level. The different levels are needed because the lower levels build the infrastructure that will be needed for the next level to be implemented successfully. For example you can't measure the effects of changes to your process(level 4), without having having a standard process(level 3). This is because it will be impossible to statistically prove what changes caused the difference in outcome, because all your projects processes differ so heavily. Grouping these different attributes into these groups allows companies to make sure they have the infrastructure they will need in order to be successful in implementing later stages. Most other groups voiced similar versions of the same idea during the whole class discussion

2.2 CMM Items Students Liked

We like the tiered list, and the clear path to built on your current maturity level. We like the flexibility of the CMM because it allowed it to be apply to whatever process my be in use at the organization.

2.3 CMM Items Students Disliked

We didn't like how this is manly geared to larger organizations, because it will make it harder to apply to our project. For example a process in stage 3 requires a team of people dedicated to the process activities, which we clearly don't have the man power for. We also didn't really like how it doesn't help if you don't have any knowledge of processes that the team could use, which makes this feel like more of a focus for middle management and difficult for a tiny team of developers to utilize in a single semester. This process also says multiple times that its a path for gradual improvement over time and the small amount of time that we will be working together will make it hard to really utilize most of the information that the CMM teaches you.

2.4 CMM in Big vs Small Companies

Since large companies have a lot of different projects going on at the same time its critically important for upper management to be able to have a good understanding of the state of all the different projects. More mature projects are easier to get this type of information on so they have a greater need for improving their processes maturity. Also having a mature process for development and management allows for the quality of work to stay consistent through staff changes. It also allows them to take improve from lessons learned in different departments so that the entire organization can improve from each other even if the individual teams don't realize it.

Smaller companies have a lot less going on so its easier for them to manage their projects even with a low level of process maturity. As they grow they will really need to start maturing their process to ensure quality and predictability throughout the increase in number of active projects. When there are less people involved at the beginning all these tools are not as necessary for management, and time could be better spend just focusing on actually delivering projects. Working on maturing the process begin using is a type of investment for the future and smaller companies may not have to resources at the start to make that investment, They might be better off just shipping product and getting paid. As their code base grows and their team grows the investment will become more worth while.

3 Our Process

3.1 Activities, Roles, and Artifacts

As of right now we don't even know what our project is, which makes me think that this assignment, or at least this question wasn't time right with the pace of the class. Since we don't yet know what we are going to be doing it really doesn't make sense for use to try and develop a process that we want to use because we are probably going to want to cater to process to fit the project, and its requirements. No one in our team has any set roles yet because we don't yet know what skills are going to be relevant to the work so its hard to assign any roles. For example if we are assigned to work on the smart CCTV system and one of us is skilled with embedded systems, we would probably want them in a more development focused role, while someone who isn't skilled with that would be given a more management focused role. We have yet to set activities but I think most of use want to use scrum because it is really designed for small teams on a strict time constraint which is exactly what we are. We will probably have a week of planning and creating the different scrum task, picking the technology and the documentation style. Then spend the rest of the semester managing our sprints.

3.2 CMM Usefulness

The most important thing for us will being using our understanding of the CMM to move as quickly as possible from level 1, to level 2. We will probably what to try a few different processes out at the start, but then decide on one and agree on the different aspects of the management technique we want to use. A big part of moving to level two will be to train ourselves on the process, and write some documentation so we are all clear on what is expected. As discussed in class CMM isn't always useful for small organizations because its an investment that over time will make projects more predictable and easier to measure. The size of our groups and the time we have to work really limits the benefit of the later levels of CMM. Chapter 4 starting by saying that even with the information from the book we aren't informed enough to carry out an assessment or evaluation also makes it harder for us to really put the CMM to practice in our projects. Basically the capability maturity model is interesting and good to know about but much of it will not be useful for the duration of this class.

3.3 Improving Maturity

The most important thing that we can do will be to get all of our technical issues out of the way at the start and develop a actionable plan of attack. This will allow us to focus on our process maturity more, giving us the time to write the documentation for our process. Once we get that done we should be able to write code to our specifications and focus on making the success of our weekly sprints more repeatable. Once we get our sprint cycle organized and under management we can then focus on getting our software process unified and under control, allowing for more uniform code between the different members.