

SOS POLITICAL SCIENCE AND
PUBLIC ADMINISTRATION

MBA FA- 206

SUBJECT NAME: FUNDAMENTALS
OF MARKETING

UNIT-III

TOPIC NAME- CONTINUOUS
IMPROVEMENT CYCLE, PRODUCT
IMPROVEMENT



CONTINUOUS IMPROVEMENT CYCLE

Continuous improvement, sometimes called continual improvement, is the ongoing improvement of products, services or processes through incremental and breakthrough improvements.

Continuous improvement is an ongoing effort to improve products, services or processes. These efforts can seek “incremental” improvement over time or “breakthrough” improvement all at once.

Among the most widely used tools for continuous improvement is a four-step quality model—the plan-do-check-act (PDCA) cycle, also known as Deming Cycle or Shewhart Cycle:

Plan: Identify an opportunity and plan for change.

Do: Implement the change on a small scale.

Check: Use data to analyze the results of the change and determine whether it made a difference.

Act: If the change was successful, implement it on a wider scale and continuously assess your results. If the change did not work, begin the cycle again.





5 EASY WAYS TO IMPROVE YOUR PRODUCT DEVELOPMENT PROCESS

Product Improvements means any invention, discovery, development or modification with respect to any Product or relating to the development, manufacture, use or commercialization of any Product, whether or not patented or patentable, including any enhancement in the efficiency, operation, manufacture, ingredients, preparation, presentation, formulation, means of delivery or dosage of any Product, any discovery or development of any new or expanded methods of treatment, use or indications for any Product or any discovery or development that improves the stability, safety or efficacy of any Product; in each case that is conceived, discovered, developed or otherwise made by Contractor, whether solely or jointly with PhaseBio, under or in connection with this Agreement or the performance of the Services hereunder.

Improving your work process doesn't have to be a huge operation if you start with small, incremental improvements. These don't require everyone in your team to read a textbook about a new methodology, or even learn new skills. These are quick improvements, which you can implement right away.

While this post is written in the context of software development, the concepts are applicable to any kind of knowledge work where you need to communicate effectively as a team, and to get tasks done as part of a collaborative effort.



1 – CREATE A WORKFLOW

This may sound straightforward, even trivial, but I've seen so many teams who haven't really thought out their basic workflow. In the most general terms, a workflow is the process that a company uses to get things done. The life and state of a task is often more complicated than simply done or not done. Usually, your tasks go through multiple stages before being completed and, in software development, there are typically phases such as planning, doing, reviewing, and deployment.

Regardless of the context of your workflow, it should be clear and explicit. Most tools allow you to customize your workflow to fit your specific methodologies so that, for example, if you want each new feature to be code reviewed, you can make it a mandatory step in your workflow. As you'll see later on in this article, it's also a very good idea to visualise your progress! It then becomes a habit like brushing your teeth – you don't need to force yourself (or someone else in your team) to do it every time.



2 – CLEAN UP YOUR BACKLOG

Filling up your backlog is easy – it's always easier to come up with new ideas than to actually implement them, but your backlog won't serve a purpose unless you are actively shaping it. A well-refined product backlog helps you to focus on what's most important right now, while not ignoring other, less critical ideas or pieces of work that have come from all manner of channels.

Chances are that your backlog contains a large amount of tasks that will never see daylight. A good practice is to split your backlog into the work that you have committed to doing in the near future, and work that you may do someday. If you use something like Trello for managing your backlog, keep these as separate lists. In fact, since you don't want low priority work to keep you distracted, I recommend having them on separate board altogether!



Clean up your backlog

The someday list consists of tasks that you may want to promote to your backlog at some point of future, if you get more detail or you start to explore a broader set of feature possibilities. It is a home for all those interesting ideas and low-priority features that you want to keep, but which do not require your immediate attention.

But don't let it become a junk yard of tasks that never get implemented! Instead, prune it periodically by deleting the tasks that no longer seem relevant or feasible. Your someday list should be somewhat stable in size, not ever expanding. When properly maintained, it can be an invaluable resource to have when choosing what to do next for your product.



3 – HAVE A CLEAR DEFINITION OF ‘DONE’

Definition of Done, or DoD, is the acceptance criteria that ensures that the tasks, once completed, are truly done, not only in terms of functionality but in terms of quality as well. Definition of Done is stated as a set of rules like:

- Feature is implemented
- Unit tests are passing
- Documentation is up-to-date
- Feature is reviewed by QA
- Code is in master branch
- Code is deployed to production



Definition of 'Done'

The Definition of Done ensures everyone on the team knows exactly what it means when something is 'done', and it removes the ambiguity from the question "are you done with this feature?" Without DoD in place, you wouldn't know whether the developer has just done the core programming part of the work, or if the feature is ready to be released for users. We've all heard someone saying "but it works on my computer..." – that's not 'Done'!

An explicit completion criteria can also reduce the ping-pong effect, which happens when QA has to repeatedly re-open the already completed task due to some unmet criteria (see step 1 above – define and refine your workflow). It's always faster to spend a little extra time and complete a task once, rather than having to revisit it several times to make fixes.

Once you've agreed on your DoD, have it prominently displayed somewhere in the team room where everyone can see it. Preferably somewhere close to the visualisation of your workflow.




4 – CONTROL YOUR WORK IN PROGRESS

(WIP)

Multitasking kills efficiency. When you overload any system, total output will go down. When you overload people with work, throughput slows and the number of defects increases. The more you multitask, the longer it takes to complete any individual task – it's a mathematical fact.

But how do you fight the creep of multitasking? To start with, you can reduce your cycle times by having fewer tasks in progress at the same time. Consider setting WIP limits to your workflow. If you're using a kanban board, you can specify the maximum number of tasks allowed on each column. For example, you can make a rule that there can't be more than two tasks in progress per developer.

It can feel counter-intuitive at first but soon you'll realise that better focus with less context switching actually leads you to complete your tasks quicker overall. In addition, you will feel better as there's less hassle and task juggling in your workday. happy workers do better quality work. it's a win-win!




5 – MAKE YOUR PROGRESS VISIBLE

Do all of your stakeholders know the current status of the project you're working on? By "stakeholders", I don't just mean your team members but also your upper management, sales people, and even the customer themselves. Is it clear to all of them which features are already completed and which ones you're currently working on? Will you be releasing on time?

A key principle of processes like kanban is to visualise your work. With a physical board on the wall (or some other so-called 'information radiator'), you walk into the room and you can immediately see what is going on. No need to distract the team by asking for status updates.

The concept of visual management dates back to the 80's when it was used by companies like Toyota for designing new car models. One room was dedicated for the design team and it's walls were covered with plans, design mockups, schedules, project progress charts – anything that people were supposed to pay attention to.



SUMMARY

The term process has picked up a negative connotation due to so many tedious and useless activities that have been associated with it. Admittedly, you can easily get too bureaucratic and top-down in its implementation, but the fact is that without some processes in place, your company will struggle as soon as you grow bigger than few employees.

Now think about the process you're using. Is it helping you to get better results? Which activities are useful, from this perspective? Are there some things that are just plain counter-productive, or that are noticeably painful and slow?

I hope you found these tips useful, let us know in the comments if you did, or if there are things you disagreed with! I'd also love to hear what other simple steps could be added to the list – what are your own tips for improving process?

