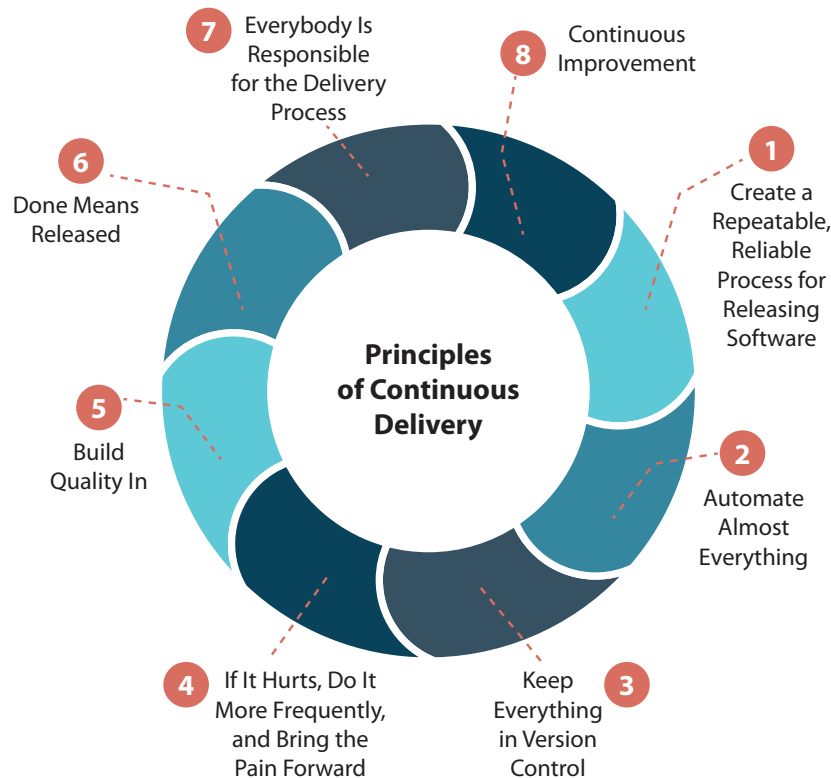


## PRINCIPLES OF CONTINUOUS DELIVERY

According to the Jolt Award Winner book, Continuous Delivery by Jez Humble and Dave Farley, there are eight principles of Continuous Delivery.



1. **Create a Repeatable, Reliable Process for Releasing Software:** When you have different release processes in different environments, you have to face several challenges to solve the raised issues whenever there is any modification in the software. You can avoid such issues by using a standard release process (as simple as pressing a button) under all environments. Repeatability and reliability depend on two principles, Automate Almost Everything and Keep Everything in Version Control.
2. **Automate Almost Everything:** You cannot consider a manual process of deploying software as repeatable and reliable. Manual processes are more prone to errors and are less efficient. An automated process requires less effort for their execution and monitoring. Therefore, spend efforts in automating repeatable and manual tasks to bring reliability and consistency.
3. **Keep Everything in Version Control:** A reliable source of truth helps you create a stable foundation to build your processes. Therefore, keep track of changes related to building, deploying, testing, and releasing an application using a version control system. It helps everyone to refer to the changesets through a single identifier, such as a build number or a version control changeset number.
4. **If It Hurts, Do It More Frequently, and Bring the Pain Forward:** Doing the same task every day, instead of doing it occasionally, helps people to become experts of that specific task and increase their productivity. Instead of ignoring or avoiding the difficult tasks, put your efforts to perform these more often. Doing the same task, again and again, helps you to improve. As a result, the same difficult task will become easy for you. Once you handle the painful areas, the rest will most likely be easier to perfect.

5. **Build Quality In:** Resolving issues later in the process is complicated and expensive. Therefore, look for issues early in the process to develop high-quality code. Feedback loops help in this direction by letting developers know about the issues as soon as the code is developed. Identifying and fixing issues during the development phase before executing the build is easier.
6. **Done Means Released:** Whenever anyone in the team says DONE, it should mean the software/product is released in production. The definition of DONE should be the same for everyone right from the start to have better communication and realization of customer value. The ownership should not be limited only to the assigned task like “It’s DONE. I have checked in the code.” So, DONE implies the ownership of the entire team for a working product in the hands of the customers.
7. **Everybody Is Responsible for the Delivery Process:** “It is working on my system. It’s good to go from my side.” Should the responsibility of team members be limited to their workstation only? Of course, not. Anything which is working on anyone’s workstation is not going to contribute to the company’s profit. Right! In the same way, if a team has no project plan, the project will never be released. If it gets released, it will never be a success. Again, no profit for the company. Therefore, responsibility should extend all the way to production. Such a cultural change is hard to implement, but not impossible. With the required support from the management and an enthusiastic champion, organizations can practice the culture of end-to-end responsibility.
8. **Continuous Improvement:** Do not stop once the product is released. There is always a scope to improve. Like practice makes a man perfect, in the same way, continuous improvement is an ongoing improvement of products, services, or processes through incremental and breakthrough improvements. It helps identify opportunities to enhance the quality of products or streamline work and remove waste from the existing process. Continuous improvement is also essential to keep your systems up-to-date with the changing technology.