

**THE CORE BELIEFS OF
WATERFALL**

**THE CORE BELIEFS OF
AGILE**

**THE CORE BELIEFS OF
LEAN**

You can know everything required to build a software product properly at the start of the project

Customer can accurately tell you what they want at the start of the project

You need to get feedback from the customer until the end of the project

Manager, developers, and customers can gauge the status of a project by looking at completed milestones as reflected in documentation. That is, given proper documentation, it is not necessary to deliver complete, tested software until the very end of the project

You can effectively have separate groups do analysis, design, code, and test. That is, there is little loss of information in the handoff between these groups

Handoffs between people in different roles can be done efficiently by writing down what was done in each step

You can test at the end of a project and achieve that required quality

Management can demand that certain work be done at a certain time and should expect it to happen

Giving people many projects to work on simultaneously is a good approach to achieving 100% productivity because then everyone is always busy.

You cannot know everything required to build a software product at the start of the project

Customers cannot accurately tell you what they want at the start of the project; instead, they will gain clarity as the project proceeds

You want feedback from customer as often as possible and you want to give developers feedback on how they are doing as soon as possible

Working code is the most accurate way of seeing the progress of the development effort

A group working together minimizes delays as well as the loss of information between people

Moving to test to the front of the development cycle improves the conversation between developers and customers and testers and, thus, improves the quality of the code

While management can set expectations for what work is done, management must not demand how that work is done

Working on one project at a time
improves the productivity of a
team

Most errors are due to the
system within which people work
rather than to the individuals
themselves.

People doing the work are the
best one to understand how to
improve the system

Ad hoc is not an acceptable
process

Looking at when things are done in a process is a more useful guide than trying to make sure every step of the way is as efficient as possible

Our measure for success must be related to the amount of time between when ideas come in and when they are manifested as value to our customer

Management must work with the team to improve the way it works to improve its own efficiency

Team are most efficient when the amount of work is limited to their capacity

Team efficiency improves by minimizing the amount of work in progress at any one time

When evaluating actions, we must optimize the whole, not merely improve individual steps in the process

There are principles in software development that must be followed in order to reduce waste