Software Development Models & Principles

Our software development processes are highly influenced by several software-development-models and frameworks.   Here is the list of models and frameworks we follow.

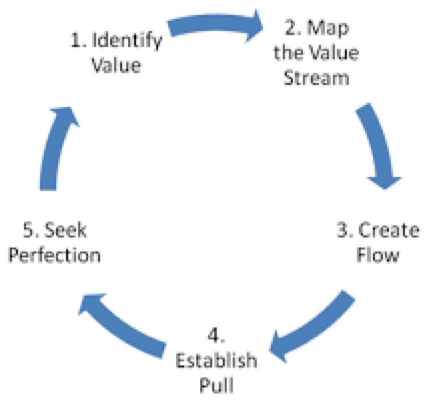
1. **LEAN Principles -** To maximize customer value with minimum resources for any business
2. **Lean Software Development** - Lean principles redefined for the software industry
3. **Agile** - A iterative software development technique, where requirements and solutions evolve through collaboration
4. **SCRUM** - Agile project management framework to do more in less time when working in a team
5. **DevOps** - The most used LEAN & Agile based framework for the digital automation of a business
6. **CMMI** - Capability / Maturity based processes to manage and control software projects efficiently
7. **Waterfall** - An non-iterative Software Development model where all requirements are defined at the start

We have an in-house end to end integrated product development system with built-in requirement management, requirement development, software modeling, defect tracking, test management, agile project management, retro management, delivery management, performance tracking, skill management system, and more. Most of the concepts and principles are automated in our integrated product development system.

LEAN Principles

These principles are more suitable for business owners than software developers. However, when we attempt to provide the business automation solutions we keep these principles in mind.

The core idea is to maximize customer value while minimizing waste. Simply, lean means creating more value for customers with fewer resources.



Step 1: Specify Value
Define value from the perspective of the final customer.

Step 2: Map
Identify all the steps in the value stream for each product family, eliminating whenever possible those

steps that do not create value.

Step 3: Flow
Make the value-creating steps occur in a tight sequence so the product will flow smoothly toward the customer.

Step 4: Pull
Let the customer pull products as needed, eliminating the need for a sales forecast.

Step 5: Perfection
There is no end to the process of reducing effort, time, space, cost, and mistakes. Return to the first step and begin the next lean transformation, offering a product that is ever more nearly what the customer wants.

How to benefit from this model?
The software application development is basically nothing but automation of business processes. And we will help you to add these values into your software application to maximize your gain with minimum resources.

Lean Software Development Principles

These are lean principles redefined for the software industry. We practice this principle for developing any software application we develop.

Principle 1: Eliminate Waste

Anything we do that does not add customer value is waste, and any delay that keeps customers from getting value when they want it is also waste. We must find out each type of waste and remove it. The biggest waste is developing unused features.

Principle 2: Build Quality In
Once a bad application is written, you can never convert it to a perfect one, however you test and fix it. Just like you can never convert a $100 phone into an iPhone, to create an iPhone Apple has added a quality built-in. We should avoid creating defects in the first place.

Principle 3: Create Knowledge
Software development is a process of turning knowledge into a software system. If the knowledge is bad, the resulting system will deliver zero value. The entire money can go to waste. The process must ensure creating the right knowledge to deliver maximum value.

Principle 4: Defer Commitment
During development, some key decisions on deciding the software knowledge are irreversible. In case we miss it, we miss it forever. So during the knowledge creation process, we should give these decisions a last chance before it is too late. Most decisions should not defer.

Principle 5: Deliver Fast
In software development slow delivery sometimes is the root of most of the problems. It simply kills the feedback loop and simply paralyzes the knowledge creation process. It is a myth that delayed delivery increases quality.

Principle 6: Respect People
Respecting people means that teams are given general plans and reasonable goals and are trusted to self-organize to meet the goals - they use their heads and figure this out for themselves. There are 2 types of people required for it. Entrepreneurial Leader and Expert Technical Workforce.

Principle 7: Optimize the Whole
A lean organization optimizes the whole value stream, from the time it receives an order to address a customer need until the software is deployed and the need is addressed. If an organization focuses on optimizing something less than the entire value stream, we can just about guarantee that the overall value stream will suffer.

How do you benefit from this model?
Software development is a process of turning knowledge into a software system. If the software development team practices these principles the resulting application has to deliver more value to its users, else it will deliver clutter. We practice these principles at the Remote Programmer.

Lean 5S

5S is a system to reduce waste and optimize productivity through maintaining an orderly workplace and using visual cues to achieve more consistent operational results. Implementation of this method "cleans up" and organizes the workplace basically in its existing configuration, and it is typically the first lean method which organizations implement. 5S pillars are

Sort -   Sort, the first S, focuses on eliminating unnecessary items from the workplace that are not needed for current production operations. Users should see relevant information and remove which are clutter. This can be done by giving the right search filter and show/hide options through proper setting.

Set in Order - Set In Order focuses on creating efficient and effective storage methods to arrange items so that they are easy to use and to label them so that they are easy to find and put away. In software, a user should see things in proper order. In the drop-down, in the menu or in the list, or at any place. The developer should be proactive enough to do it in the first place.

Shine - The workplace should look clean and devoid of clutter. In software, the workplaces are dashboards and different screens.

Standardize - Once the first three 5S's have been implemented, the next pillar is to standardize the best practices to be reused. A user should not face different experiences for similar action.

Sustain - Maintain the above 4S principles over time.

How do you benefit from this model?
There is a dramatic effect on the user experience if 5S is implemented properly in your application. It will reduce user time, boredom, and irritation. We practice it at the Remote Programmer.

Agile Manifesto: 4 Key Values

Value 1: Individuals and interactions
In the past, a lot of software teams would concentrate on having the best possible tools or processes with which to build their software. The Agile Manifesto suggests that while those things are important, the people behind the processes are even more so.

Value 2: Working software
Previously, software developers would spend ages creating detailed documentation. That was before they even started writing a single line of code. And while documentation isn't a bad thing, there comes a point when you should focus on providing your customers with working software.

Value 3: Customer collaboration
Once upon a time, contracts were king. You would draw up contracts with your customers who would then detail the finished product. As a result, there was often a contrast between what the contract said, what the product did, and what the customer actually required.

Value 4. Responding to change over following a plan
Traditional software development regarded change as an expense, so it was to be avoided. But in agile change is the first priority over following a plan.

How do you benefit from this model?
As per software dynamics, 20% of features provides 80% values. So if we can deliver this 20% perfectly you can quickly reach your customers/users with 80% values. In the next version 20% of the rest, 16% of the total. So 36% of the development will result in 96% values. You will be automatically ahead of your competitor. But choosing 20% and building it perfectly is important. We will help you by using the agile method.

The DevOps 3 Ways

As per the DevOps framework Development team and operation team, stops blaming each other and work hand in hand. The full organization is treated as a single system and the software development team is not considered as a third party. DevOps is basically the successor of agile and Lean. It is based on 3 key principles. It is a default framework for the full digitalization of an organization.

The first way: Flow (System thinking)
The First Way emphasizes the performance of the entire system, as opposed to the performance of a specific silo of work or department. The idea behind Systems Thinking is to move the product as quickly as possible from left to right, from the developers to the operational. Ideally, the system should cover all the functional areas and remove all possible manual work by proper automation of the workflow.

The 2nd Way: Amplify the Feedback Loop
The Second Way is about creating the right to left feedback loops. The goal of almost any process improvement initiative is to shorten and amplify feedback loops so necessary corrections can be continually made.

The outcomes of the Second Way include understanding and responding to all customers, internal and external, shortening and amplifying all feedback loops, and embedding knowledge where we need it.

The 3rd way: Continual Experimentation and Learning
The Third Way is about creating a culture that fosters two things: continual experimentation, taking risks and learning from failure; and understanding that repetition and practice are the prerequisites to mastery.

How do you benefit from this model?
To benefit from it you need to make the development team a part of your business operation. The Dev team will first automate its own product development and project management processes to make the development process transparent to everyone in the team. Then it will automate the business processes one by one. By developing and/or integrating features / modules / applications. Then constantly improving the whole system to maximize the flow and feedback.

CMMI Process Areas

The agile or lean sets high-level principles or frameworks but it does not provide any specific methodology - how software should be developed. CMMI sets specific process areas to do software development works.

In fact, we try to follow 21 of the CMMI process areas. We automate many of them through DevOps first way. Here are the CMMI processes areas we maintain (non-greyed ones).

Maturity Level 1 - No defined process

Maturity Level 2 - Managed

- CM - Configuration Management
- MA - Measurement and Analysis
- PMC - Project Monitoring and Control
- PP - Project Planning
- PPQA - Process and Product Quality Assurance
- REQM - Requirements Management
- SAM - Supplier Agreement Management

Maturity Level 3 - Defined

- DAR - Decision Analysis and Resolution
- IPM - Integrated Project Management
- OPD - Organizational Process Definition
- OPF - Organizational Process Focus
- OT - Organizational Training
- PI - Product Integration
- RD - Requirements Development
- RSKM - Risk Management
- TS - Technical Solution

- VAL - Validation
- VER - Verification

Maturity Level 4 - Quantitatively Managed

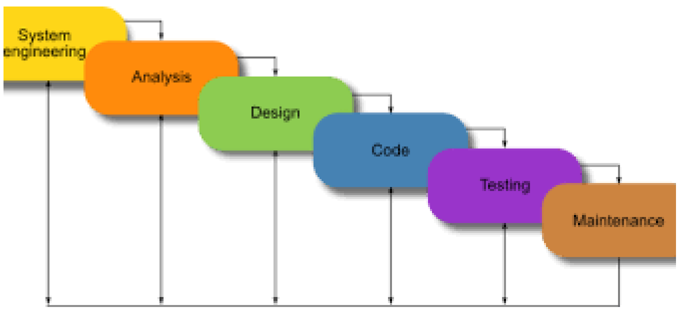
- OPP - Organizational Process Performance
- QPM - Quantitative Project Management

Maturity Level 5 - Optimizing

- CAR - Causal Analysis and Resolution
- OPM - Organizational Performance Management

How do you benefit from this model?
These are basically internal processes of software development. We simply practice these processes during software development. Most of the process areas are automated in our agile product cum project management software. So you get the benefit automatically.

Waterfall SDLC Model



This is a traditional software development model and it is still relevant for doing fixed requirement small projects where client involvement is not required in the middle of the project. It assumes that all requirements can be defined at the start of the project and most decisions on development and funding are taken at the start. For a change, a new deal needs to be made.

How do you benefit from this model?
This model should not be used in software product development where all requirements can not be defined at the start. However, in many cases, the stake owners are not during development, in those situations, development can be done in a mini waterfall model. That means the full applications are divided into subsystems and each subsystem is to be developed following the waterfall model.