

Flutter In Action: Software Delivery Methods : Difficult yet easy choice

Posted by [Khushboo Uchat](#) Feb 3, 2021

From the title you might not choose to read ahead as it is same old boring topic but let me tell you that I will keep it short and interesting. You have guessed it right what this blog is about. So lets jump right in the topic.

We all are working in software industry and delivering software is one of the task which we perform almost daily. But have you ever thought that **the way you are delivering software** makes the lots of difference though end product can be same? **It is just like eating the food.** The end result is same that you are eating it all but what if someone tells you to eat desert before main course and soup at last (Bless my imagination..). There are several to choose from and each has its own pros and cons

Before going in various methods, I would like to talk about one interesting **survey about Agile conducted in 2012 by VersionOne** who polled 4,048 individuals from various channels. The results were interesting and eye opener.

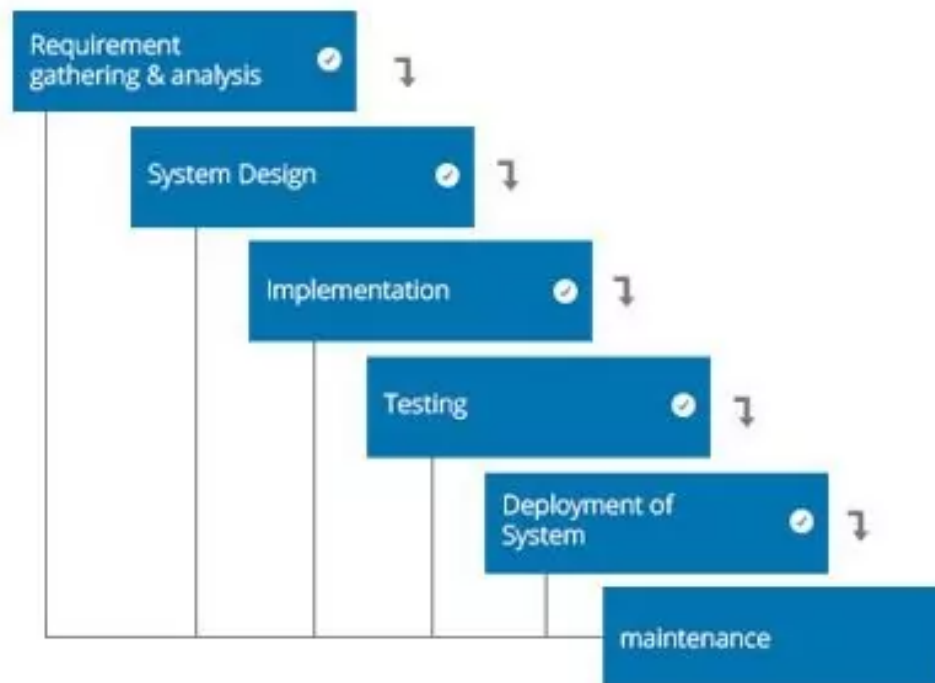
- When asked about **who knows Agile most** then most of them **said that the roles which are near to the team knows it most than the business roles.**
- When asked about **barrier to adopt Agile** then **53% said that disability to change org culture**
- When asked about **greatest concern while adopting Agile** then **43% said lack of upfront planning is the reason.**

So if you are also in dilemma then this blog post might help you to choose right development method for your project

1. Waterfall Model

It is the pioneer of SDLC and parent of all other methods. It was the first widely used model in software industry and other methods are invented due to the lack of features in either of the phase of waterfall.

It is example of the sequential model where the development is broken in series of steps and at each phase the objectives are different. It is necessary to complete the current phase before going to next one.



Use Case : It may seem useless in current world of agile but waterfall can be proven advantageous **when your requirement is stable and your application is very small.**

Pros & Cons: Due to lack of feedback at each stage if changes are to be incorporated then it can prove expensive on the other hand for small applications it provides clarity with less effort.

My Experience with Waterfall

In 2011, I was working in the start up based on PHP and the **end products was set of php webpages or small website based on php and mySQL**. I used to follow waterfall model as the requirement from the start was very clear and no end user was immediately available to provide feedback. It provided a very good understanding and clarity of importance of each phase and established a base on which I can evaluated other methods of development.

2. RAD (Rapid Application Development)

This method was born out of frustration of waterfall development system. It is **a concept which emphasizes working on software and being more adaptive than older development methods**. The term 'Rapid' was born when James Martin was trying to develop a new method

called Rapid Iterative Production Prototyping (RIPP). As the name suggests, it is believed that software can be delivered faster by

- Gathering requirements using workshops or focus groups
- Prototyping and early, reiterative user testing of designs
- The re-use of software components

Use Case : RAD mostly goes with fully OOP supportive languages like C++ or Java because RAD may fail where clear modularization is not possible as this model supports incremental deliveries only.

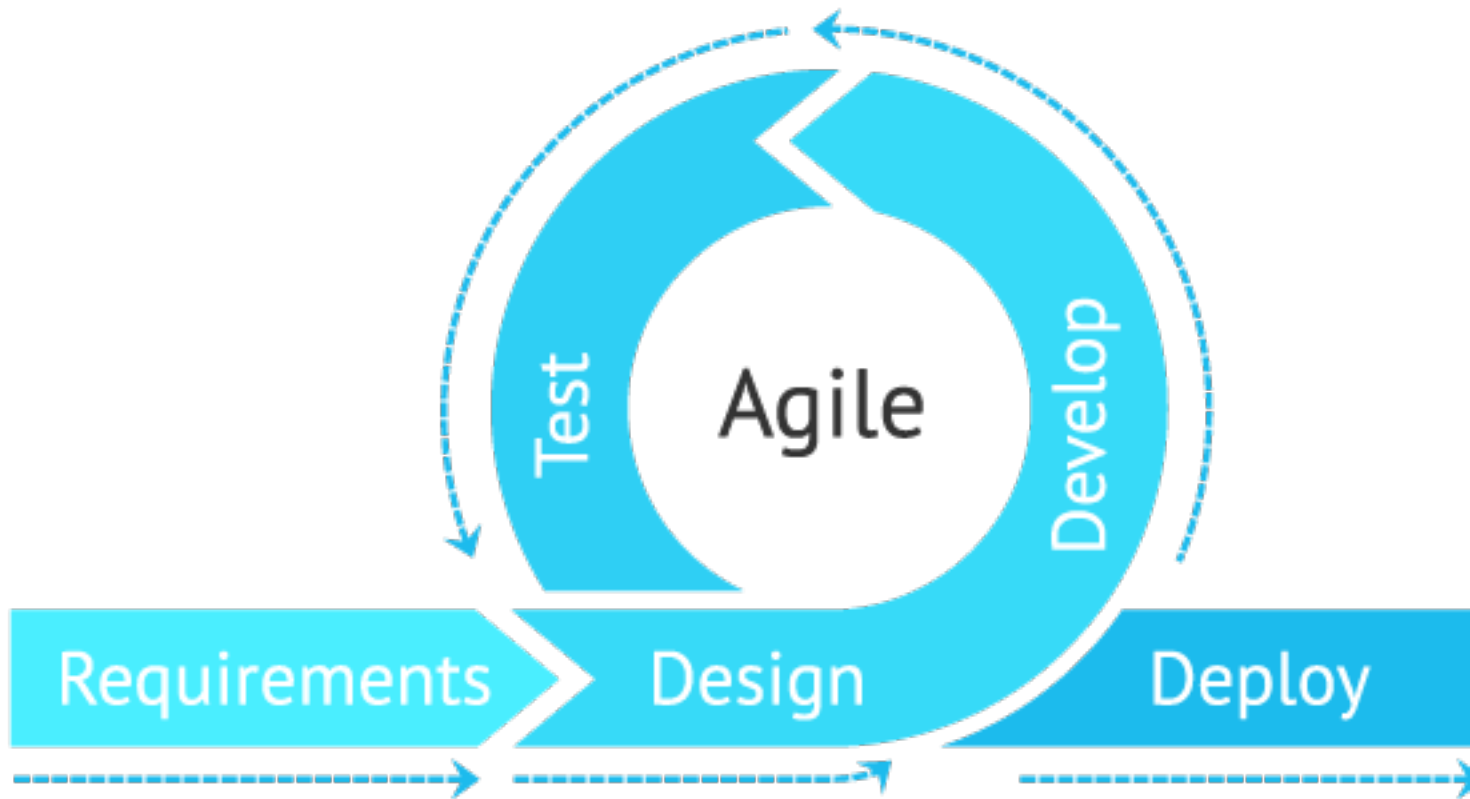
Pros & Cons : RAD can give short iterations and faster deliveries with powerful RAD tools but cannot be used where modularization fails and clear boundaries are not defined.

My Experience with RAD

In Deloitte during my career, I have worked as Java + Angular developer. Our web application was built on Spring with RAD approach. The plus point of RAD has that it has boilerplate code ready and you need not worry about that part during the whole SDLC as it is already matured before you start your actual development. You just need to worry about customization part and in this way you can deliver the code faster. We had our boilerplate build on AEM and customization done using java with clear modularization visible.

3. Agile

Now we come to the method which is talk of the town now a days. Everyone seems to be fascinated by the word 'Agile' but lots of them does not know what that exactly means. In a few words, it means 'coping up with changing requirement' while developing software before delivering final product. There is no clear definition of Agile and different team adopts to its own Agile based on various parameters.



Agile Manifesto

After seeing issues in waterfall for the years, on February 11-13, 2001 seventeen people met to talk, ski, relax, and try to find common ground and what emerged is called 'Agile manifesto'. It says

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

- *Individuals and interactions* over processes and tools
- *Working software* over *comprehensive documentation*
- *Customer collaboration* over contract negotiation
- *Responding to change* over following a plan

Waterfall vs Agile

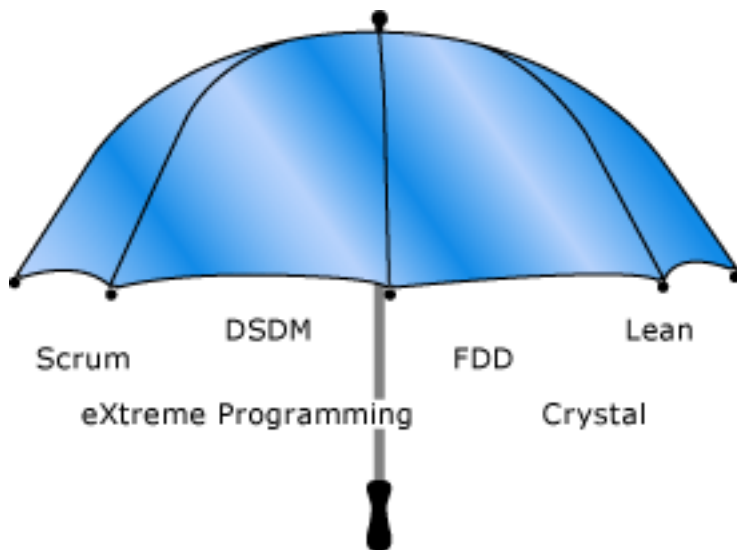
From the above diagram it is quite visible that Agile gives solutions to lots of issues started with waterfall programming.

Pros & Cons : the quality of the outcome improves as testing starts parallel to development and risk is reduced due to finding of bugs in early stage of development. It also addresses the needs of rapidly changing requirement by providing feedback at each stage via iterations. As per principles the documentation is not

considered important, sometimes it becomes the bottleneck when large team is dissolved or the key person leaves the team.

Umbrella of Agile

As I mentioned, Agile is not the method itself but it is umbrella term and every one has its own version of Agile which basically satisfies the above criteria and can serve to other purposes as well. Once any team starts to use Agile they might find themselves following either one of the below or mixture of two or more forms of Agile.



3.1 DSDM (Dynamic software development method)

The DSDM (Dynamic Software Development Method) was developed to fill in some of the gaps in the RAD method by providing a framework which takes into account the entire development cycle. RAD became popular with the lack of structure and to provide some structure, In 1994, DSDM is found with the focus on MoSCoW technique to prioritize the issues. MoSCoW stand for

- MUST have
- SHOULD have
- COULD have
- WON'T have

3.2 XP (Extreme Programming)

According to Wiki definition: "Extreme Programming (XP) is a software development methodology which is intended to improve software quality and responsiveness to changing customer requirements. As a type of agile software development, it advocates frequent "releases" in short development cycles, which is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted."

In simple words, it focuses on multiple releases in a day may be but one release can be equal to one line of code which is considered smallest unbreakable change. This methodology is only possible to adopt when you have automated release and CI/CD full fledged integrated with your environment. For e.g Amazon releases the code every 11.7 seconds. This is the best example where you see the actual XP in action.

Pros & Cons : This can be advantageous only in fully automated devops and CI/CD in place and makes testing easy as we are releasing smallest change which is easily testable.

My Experience with XP

In my previous organization during 2018-2020 I have been part of the team working on Traders Dashboard and that dashboard was very high visibility product internal to organization. I have done 29 releases in a week at best where release can contain smallest change and need not be huge code.

3.3 Scrum

Its focus is on "a flexible, holistic product development strategy where a development team works as a unit to reach a common goal" as opposed to a "traditional, sequential approach". Scrum focuses on why does it take so long and so much effort to do stuff and why are we not able to figure out how long and how much effort things will take. Oriflame

There are few predefined roles which is the core of 'Scrum' and which drives the scrum. The roles are

1. Product Owner
2. Development Team
3. ScrumMaster
4. *Stakeholders*
5. *Managers*

Pros & Cons: It can help achieve large deliveries as it can easily be divided in sprints and team gets clarity of issues during scrum meetings so communication gaps are avoided efficiently. But the chances of failures are quite high if individual is not committed and there is no definite end-rate.

My experience with Scrum

Currently I am part of the Arena team where we follow scrum method as Agile. As mentioned above, we try to achieve the common goal which is to make the product better for the user by providing high priority features and facilitating the user to achieve the task with ease.

There are lots of other methods like TDD, BDD and Lean but one blog might not be sufficient to discuss on them so I might write about them on other blog post. Please share your thoughts if you like this blog post and helped you identify the right method for you.

Flutter In Action: Software Delivery Methods : Difficult yet easy choice

34 Views Tags: waterfall, extreme_programming, be agile, software development methodologies, agile myth, agile vs. waterfall



[Nihar Qamar](#)

Mar 8, 2021 12:07 PM

lovely overview, thank you for putting together!