How did people develop Software about 50 to 70 years go.

About 30 to 40 years ago, US military said “Predictive process”

We write software because no one else has written this code.

Complexity:

Building a nuclear reactor

Build a new feature in software

Death March.

HW is of good quality. SW is not of same standard.

Successful: On time, On budget

Failed: Death March

Challenged: time exceeded, budget exceeded.

YAGNI: You ain’t gonna need it.

Microsoft observed how a book writer uses Word.

Root cause of YAGNI is Predictive process.

Most problems are process related, not people related.

People not having technical skills. – Process related.

Changing process is difficult, time consuming.

Simple Agile Project:

* Everyone co-located.
* Small team.
* Greenfield project.
* T&M

Iteration 0: 2 weeks.

1. Vision document
2. High level architecture
3. Product backlog: User stories priortized. E.g. 200 US.

Iteration 1: 2 weeks.

Pick 5 US. For these 5 US: Analysis, Design, Coding, Testing.

Iteration 2: 2 weeks.

Pick 6 US. For these 6 US: Analysis, Design, Coding, Unit testing.

For 11 US: Integration testing, System testing.

Iteration 3: 2 weeks.

Pick 4 US: For these 4 US: Analysis, Design, Coding, Unit Testing.

For 15 US: Integration testing, System testing.

20% features provide 80% value.

Option 1: Top is Predictive, Bottom is Agile.

Option 2: Top is Agile, Bottom is Predictive. Better

Predictive: Project will take 12 months.

Agile: After 2 months, they could start selling.

Compare Predictive with Agile:

* For long projects, risk is lower Agile.
* Lesser overhead in Agile.
* Faster feedback in Agile.
* Time to market is faster.
* Resource planning is easier in Agile.
* Changes are simpler and faster in Agile.

Cycle Time: Duration from the customer saying this is the most important feature, to putting the feature in production.

Predictive is marketed as fixed cost and fixed time.

Agile can be executed as fixed cost and fixed time; provided you are willing to adjust scope.

No documentation is similar to huge documentation.

Typical Agile project: 100 pages of documenation. Document like how to build and crucial parts of the project.

No YAGNI, No huge docs

When to use:

Code & Fix: We use for small projects (less than 2 months) with 1 or 2 team members.

Predictive: When all the following conditions are met

1. Requriements are clear and won’t change.
2. All features are needed for software to be used.

E.g Chadrayan-3, Election

Most projects, you are better off with agile.

Product Owner – Car driver

Team - Car

ScrumMaster - Car Technician.

Predictive: One development team; one Testing team.

Agile: One team

Who wants to be a millionarie?

* 50/50. 0.5
* Phone a friend. 0.65
* Audience poll. 0.91