# Agile Software Development

⚠ Waterfall Model phases

- requirements
- analysis and design
- development
- testing
- deployment and maintenance

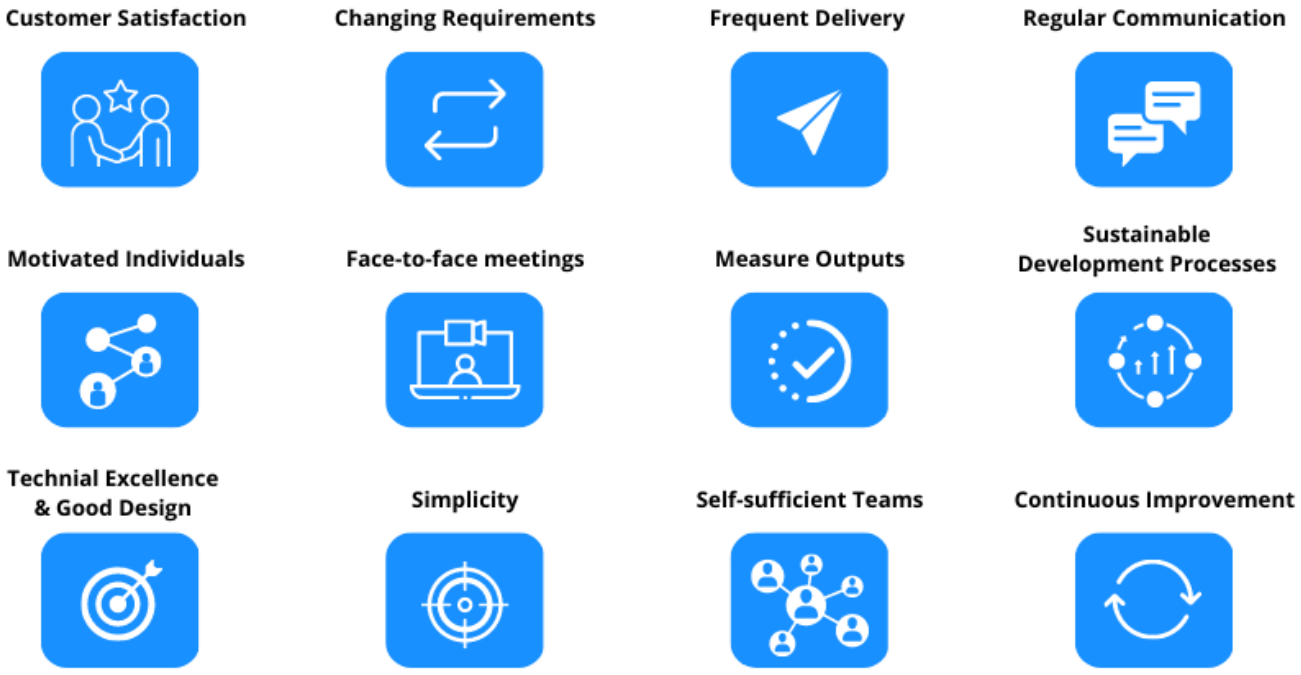
⚠ Waterfall issues

- uses top-down management
- it could be a lot of waste of time and budget (Customer will see the product at 4th level)

⚠ Agile Manifesto

- indivisuals and interactions over processes & tools
- working software over comprehensive documentation
- customer collaboration over contract negotiation
- responding to change over following a plan

# Agile Principles




| | | |
|---|---|---|
| 01 Our highest priority is to satisfy the customer through early and continuous delivery of valuable software. | 02 Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage. | 03 Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale. |
| 04 Business people and developers must work together daily throughout the project. | 05 Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done. | 06 Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely. |
| 07 Working software is the primary measure of progress. | 08 The most efficient and effective method of conveying information to and within a development team is face-to-face conversation. | 09 Continuous attention to technical excellence and good design enhances agility. |
| 10 Simplicity – the art of maximizing the amount of work not done – is essential. | 11 The best architectures, requirements, and designs emerge from self-organizing teams. | 12 At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly. |

⚠ Scrum

↳ A framework that help team build complex teams

⚠ Sprint goal is the high-level goal of each time-boxed sprint written as concisely as possible

⚠ Scrum events

- sprint planning → everyone has its own time-box
- sprint daily stand up
- sprint retrospective
- sprint review

+ sprint

⚠ 3 pillars of scrum

- transparency
- inspection
- adaptation

⚠ Scrum values

- focus
- respect
- openness
- courage
- commitment

⚠ product owner decides what to be built

⚠ $3 \leq$ development team size $\leq 9$

⚠ Scrum master facilitate processes by addressing impediments and obstacles

⚠ only product owner can cancel the sprint

⚠ Scrum artifacts

- product backlog
- sprint backlog
- product increment

⚠ sprint grooming is not officialy an scrum event

⚠ feature teams

- cross-functional
- ability to implement end-to-end functionality

⚠ component teams

- specialized around specific components
- would require multiple component teams to implement end-to-end functionality

⚠ Daily meeting is not a status meeting

⚠ XP has weekly and quaterly cycles

↳ use pair programming & continous integration

⚠ Pair programming has some benefits like knowledge sharing and immediate feedback

⚠ Test driven development

↳ write a test that fails

↳ write enough code that compile

↳ complete code to meet requirements of test

⚠ Code coverage should be included in DoD document

⚠ User stories should be written from user's point of view

⚠ Epic is much larger than an user story

⚠ Two common series for estimation

↳ Fibonacci

↳ exponential

↳ another method : T-shirt sizing

⚠ Typical deployment

↳ ① Business ② Development ③ QA

④ IT operations ⑤ Production

⚠ Dev-Ops

↳ Continous Integration (CI)

↳ practice of freqeuntly commiting changes to the source code

↳ Continous Delivery (CD)

↳ capability to keep product stable after every change so that it's always shippable

⚠ Burn down chart

↳ amount of remaining work ⟶ Y-Axis

↳ sprint time ⟶ X-Axis

⚠ Little's Law for software development ( $L = \lambda W$ )

↳ Work in Progress = Completion Rate × cycle time

$L$ $\lambda$ $W$

⚠ kanban typical steps

- planning / coordination
- design
- develop
- accept

⚠ kanban has WIP (Work In Progress) limit for each step

usually → Limit = number of team members + buffer

⚠ kanban is a process management tool not a framework

⚠ kanban is more lightweight than scrum

⚠ Nexus

- essentially scrum with one additional artifact, a few additional events, and one additional team role
- principles of scrum also apply to nexus
- execution is like running multiple scrum teams in parallel that produce one integrated product increment at end of sprint

⚠ Another scaling scrum frameworks

- SOSOS
- LeSS
- SAFe