Waterfall method follows very simple pattern. At first all the requirements are gathered. Then final product is designed, and the product development process may begin. After product is finished, it is tested. Finally, product is released and only jobs left to do are maintenance jobs.

Unified Process is iterative and incremental, use-case driven and architecture centric software development method. At first, all the requirements are gathered. After gathering requirements, use-cases are created along with other diagrams like domain model, sequence diagrams, communication diagrams and design class diagram. Work is done in iterations, iterations have 4 phases- Inception, elaboration, construction and transition. Tests are usually done at the end of each iteration.

Plan driven development

Pros

Clear overview of project

Documentation is easy to write

Perfect for small projects, in which requirements do not change

Good for beginners – tasks are known from the beginning

Cons

Changes are hard to implement

Little communication between customer and team

Little time for fixing errors

Extreme Programming (XP) is an agile planning and development method. It is based on 4 values (communication, simplicity, feedback and courage) and 12 principles (Planning Game, Small releases, Metaphor, Simple design, Define test first, Refactoring, Pair programming, Collective ownership, Continuous integration, 37 hour week, On-site customer, Coding standards). XP methodology starts with planning game. User stories are created to describe functions in the program. Then they must go through acceptance test. Next up is estimations for user stories. Team estimates time it takes to finish a certain user story, block of user stories or even the whole project. A good idea is to create burndown chart as well to visualize pogress in project’s development. XP implements some coding practises like pair programming and test driven development (TDD). Pair programming is a clever way to solve problems about complex parts of project. TDD is smart way to prevent code from stopping to do what it is supposed to do. For example, when implementing modifications to existing code, one or more functions might get lost. TDD helps keep track of all the functions in the project by forcing one to create test before writing actual method.

Scrum is an agile planning and controlling method. Scrum has 3 roles (product owner, scrum master, team), 3 ceremonies (sprint planning, daily meeting, sprint retrospective) and 3 artifacts (product backlog, sprint backlog, burndown chart). At first the scrum master and product owner set up product backlog. Items are then prioritized and time estimations are added. After time estimations are added, it is possible to create burndown chart. Items are then moved from product backlog into sprint backlog and sprint can begin. Daily meeting are also part of scrum, they are conducted by Scrum master. Meetings are used to get an overview of the state in which the project currently is. People discuss what were they doing last time (in last meeting), what will they do this time and if there are any problems. Working in iterations is useful, because at the end of each sprint the team presents a working product. Working product shows clearly what has been done in previous sprint, it makes doing sprint retrospective easier. If changes in the requirements occur, then new tasks can be added to product backlog at any time and from there they can be put into sprint backlog for development.

Kanban is agile development method which uses kanban board for deviding workflow. Kanban has 3 rules: visualize workflow, put limit to amount of items in work in progress area and esitmate time it would take to finish the task. Kanban has fewer rules than scrum and therefore gives more freedom for the developer. Kanban uses user stories, they are put up on a board into backlog section. Then items are taken from backlog area and put into selected area (acts as sprint backlog). From there developer takes a task and puts it into development section, which means that this user story is now in development. After finishing with this user story, it is put into testing area. If all tests give expected results, then the user story is moved into final, live area, meaning that it is finished and ready to be released. Limit on work in progress area (development area) is important, because as soon as development of ne part of the project stops or slows down, it is visible on the board. WIP area will be filled with only assignments from one part of project and since no more assignments can be added, then the issue must be resolved in order to continue with everything else. Kanban has the best solution for dealing with changes in program. New user stories can be put into backlog area on the board at any time.

Agile

Pros

Changes are easy to implement

Incremental delivery

Continious testing, not only at the end of development process

More communication between product owner and team

More freedom for developer

Cons

Big fixed price projects are difficult to manage

Prioritizing user stories can be a challenge with multiple product owners

Refactoring is expensive