|  |  |
| --- | --- |
| **Agile Method** | **Features** |
| Extreme Programming | 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.  Highlighted Features:   1. Metaphor - Guide development with simple shared story of how the whole system works. 2. Refactoring - Restructure system without changing its behavior to remove duplication, improve communication, add flexibility and simplify. 3. Pair Programming - Two programmers, one machine, four eyes are better than two. 4. Collective Ownership - Anyone can change code anywhere in the system at any time. 5. Continuous Integration - Integrate and build the system many times a day, every time a talk is completed. 6. 40-hour Week - Never work overtime a second week in a row. 7. On-site Customer - Real, live user on the team, available full-time to answer questions. 8. Create spike solutions to figure out answers to tough technical or design problems. A spike solution is a very simple program to explore potential solutions. Build the spike to only addresses the problem under examination and ignore all other concerns. |
| Lean Software Development | Lean Software Development is more strategically focused than other Agile methodology. The goals are to develop software in one-third the time, with one-third the budget, and with one-third the defect rate.  Highlighted Features:   1. **Eliminate waste -** In software development, waste is anything that does not improve the quality of code, reduces the amount of time and effort it takes to produce code, or does not deliver business value to the customer. 2. **Amplify Learning** - For programmers to develop a system that delivers business value, they will have to learn about many things. 3. **Decide as late as possible** - The idea here is to wait until what the authors term “the last responsible moment” to make a decision. 4. **Deliver as fast as possible** -Typical 9-12 month projects generate roughly a 25 percent change in requirements. However, the amount of requirements change over a month averages only 1-2 percent. And it is much easier to get users to accept waiting until next month rather than next year. 5. **Empower the team** -In order to get people to take responsibility, get motivated, and gel as a team, they need to be responsible for the outcome and authorized to make it happen. 6. **Build integrity** -Conceptual integrity is how well the architecture and system components flow together to bring about the perceived integrity. Obviously testing, unit and integration, is a major part of integrity. 7. **See the whole** -Systems thinking has been around for a while, but the typical response to solving problems is to break them down into their constituent parts and optimize each individual piece. |
|  | The word Kan means "visual" in Japanese and the word "ban" means "card". So Kanban refers to "visual cards“.  **Highlighted Features**   1. identify and eliminate bottlenecks and achieve dramatic operational improvements in terms of throughput and quality 2. reduced lead time, increased throughput and much higher quality of products or services delivered |