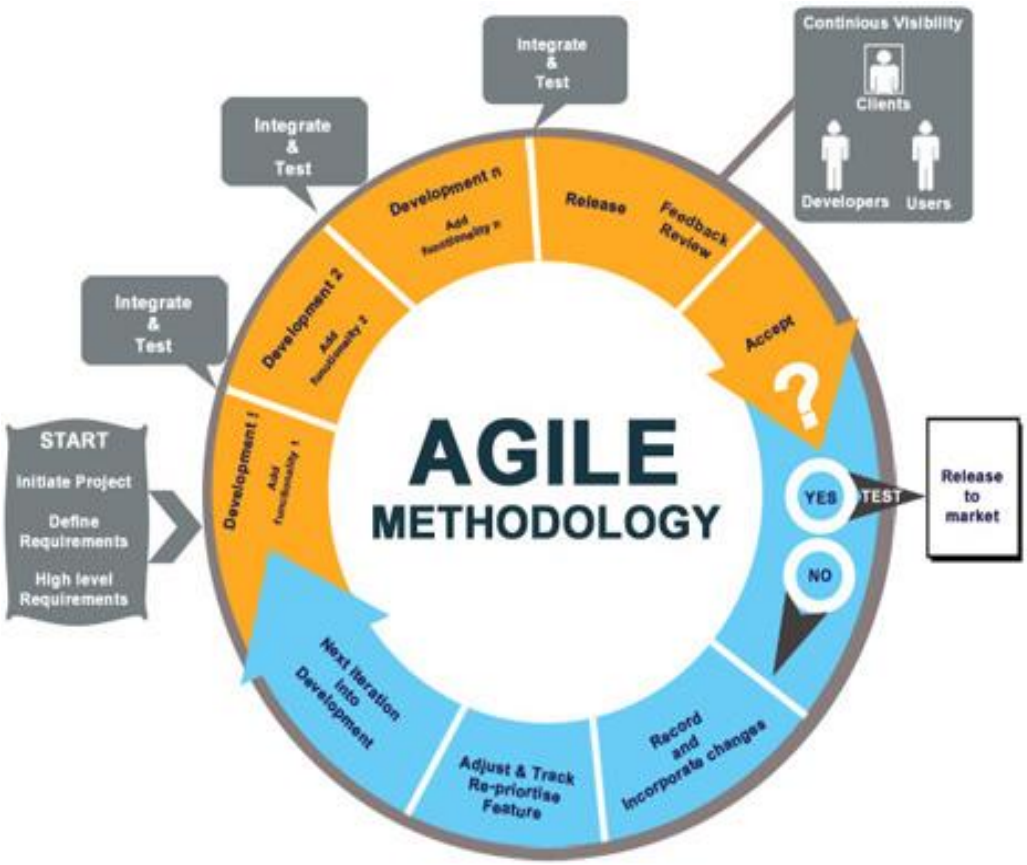
Assignment 2

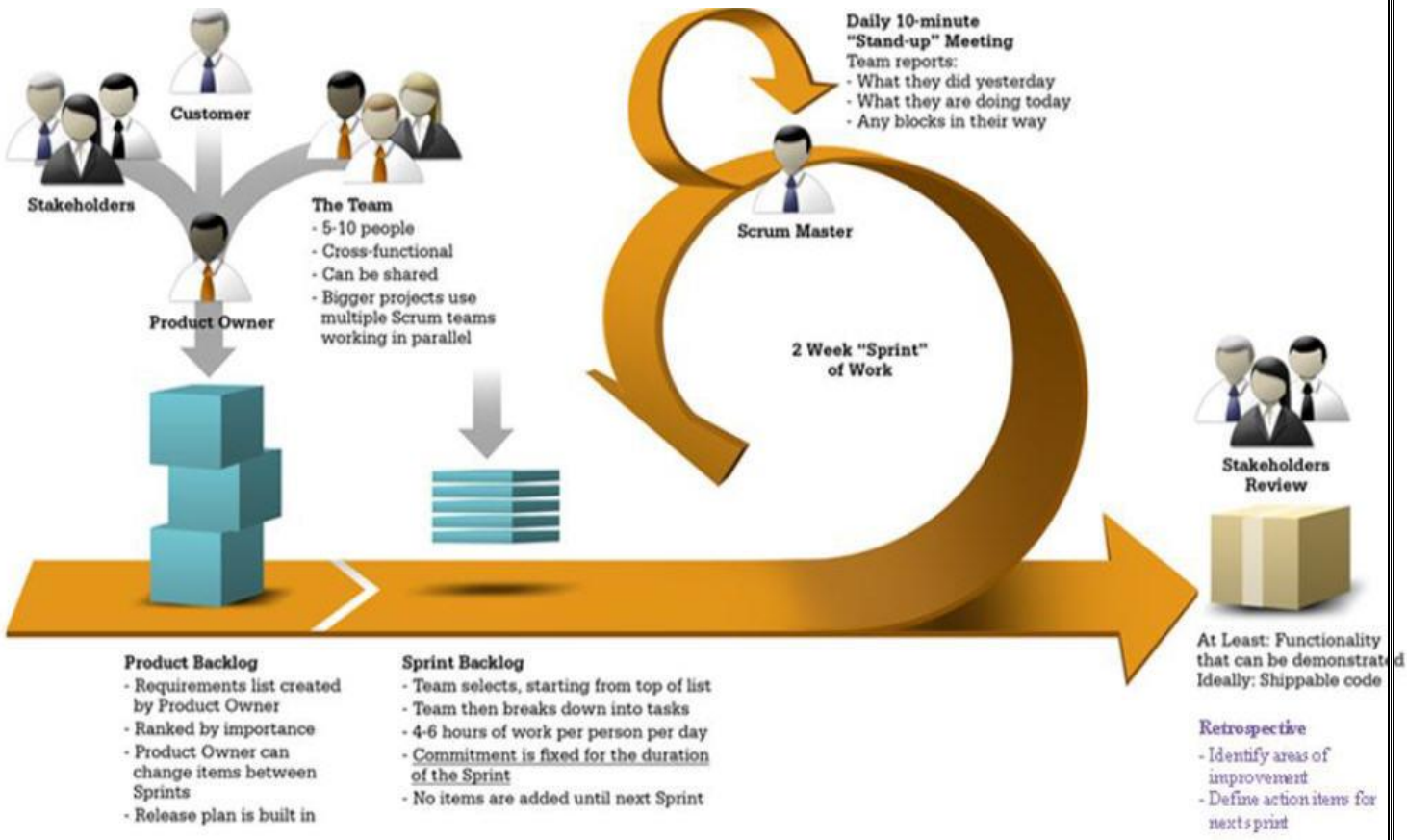
Software Development Life Cycle: - Agile

Agile development model is also a type of Incremental model. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly tested to ensure software quality is maintained. It is used for time critical applications. Extreme Programming (XP) is currently one of the most well-known agile development life cycle model.



Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.



Agile Processes

- Are based on three key assumptions
    1. It is difficult to predict in advance which requirements or customer priorities will change and which will not
    2. For many types of software design and construction activities are interleaved (construction is used to prove the design)
    3. Analysis, design, and testing are not as predictable from a planning perspective as one might like them to be
- Agile processes must be adapted incrementally to manage unpredictability
- Incremental adaptation requires customer feedback based on evaluation of delivered software increments (executable prototypes) over short time periods.

Agility Principles

- Highest priority is to satisfy customer through early and continuous delivery of valuable software
- Welcome changing requirements even late in development, accommodating change is viewed as increasing the customer's competitive advantage
- Delivering working software frequently with a preference for shorter delivery schedules (e.g., every 2 or 3 weeks)
- Business people and developers must work together daily during the project
- Build projects around motivated individuals, given them the environment and support they need, trust them to get the job done
- Face-to-face communication is the most effective method of conveying information within the development team
- Working software is the primary measure of progress
- Agile processes support sustainable development, developers and customers should be able to continue development indefinitely
- Continuous attention to technical excellence and good design enhances agility
- Simplicity (defined as maximizing the work not done) is essential
- The best architectures, requirements, and design emerge from self-organizing teams
- At regular intervals teams reflects how to become more effective and adjusts its behaviour accordingly

Human Factors

- Traits that need to exist in members of agile development teams:
    - Competence
    - Common focus
    - Collaboration
    - Decision-making ability
    - Fuzzy-problem solving ability
    - Mutual trust and respect
    - Self-organization

Advantages of Agile model:

- Customer satisfaction by rapid, continuous delivery of useful software.
- People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
- Working software is delivered frequently (weeks rather than months).
- Face-to-face conversation is the best form of communication.
- Close, daily cooperation between business people and developers.
- Continuous attention to technical excellence and good design.
- Regular adaptation to changing circumstances.
- Even late changes in requirements are welcomed

Disadvantages of Agile model:

- In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- There is lack of emphasis on necessary designing and documentation.
- The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
- Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

When to use agile model:

- When new changes are needed to be implemented. The freedom agile gives to change is very important. New changes can be implemented at very little cost because of the frequency of new increments that are produced.
- To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.
- Unlike the waterfall model in agile model very limited planning is required to get started with the project. Agile assumes that the end users' needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be newly effected or removed based on feedback. This effectively gives the customer the finished system they want or need.
- Both system developers and stakeholders alike, find they also get more freedom of time and options than if the software was developed in a more rigid sequential way. Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill.