CSE307: Software Engineering

Process Models and Agile Development

Process Models

Software Development - Social Learning Process

- A software is basically an embodied knowledge that is initially scattered and incomplete.
- We need to follow a step-by-step process to bring knowledge into a software.
- Software development is essentially an iterative social learning process.

What is Process Model?

- Software development is a collection of different activities.
- A process model basically is a technique to divide these software development works into distinct phases.
- It provides a visual road map to organize and improve both software project and software product management.

What is Process Model?

Generic Process Models

- Generic process model provides definitive description of process in a software project.

- There are 5 (five) core activities (aka framework activities) defined in a generic process model.

  1. **Communication:** Project team communicates with client.

- There are 5 (five) core activities (aka framework activities) defined in a generic process model.

  2. <u>Planning:</u> Project team schedules the project development and estimates the project cost.

  3. <u>Modeling:</u> Project team analyzes the client requirements and designs the algorithm and flowchart of different software components.

- There are 5 (five) core activities (aka framework activities) defined in a generic process model.

  4. <u>Construction:</u> Project team implements and tests different software components.

  5. <u>Deployment:</u> Project team delivers the software product to client and takes feedback to improve the quality of the product if required.

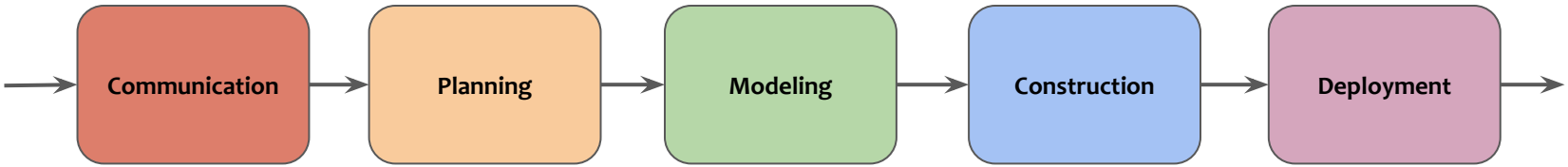
Generic Process Models

- There are 4 (four) unique process flows in generic process model.

1. Linear Process Flow
2. Iterative Process Flow
3. Evolutionary Process Flow
4. Parallel Process Flow

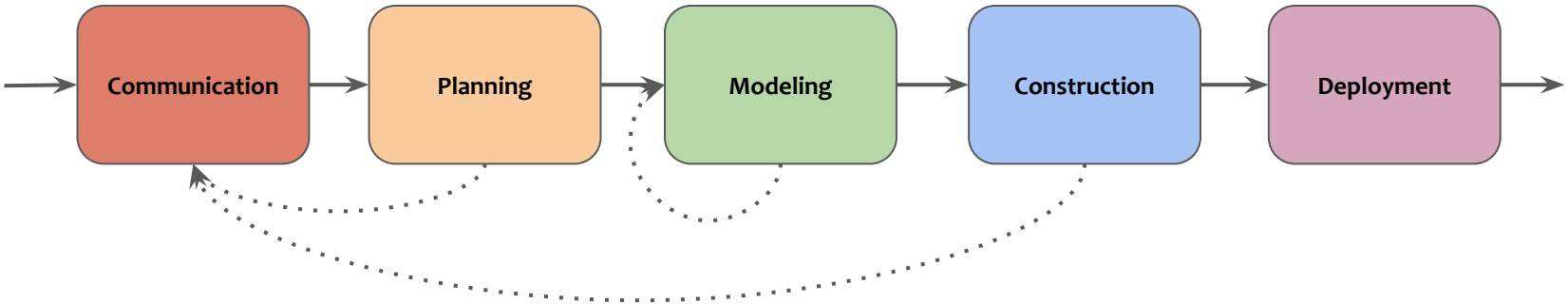
Linear Process Flow

- Linear process flow executes each of the 5 (five) activities sequentially.



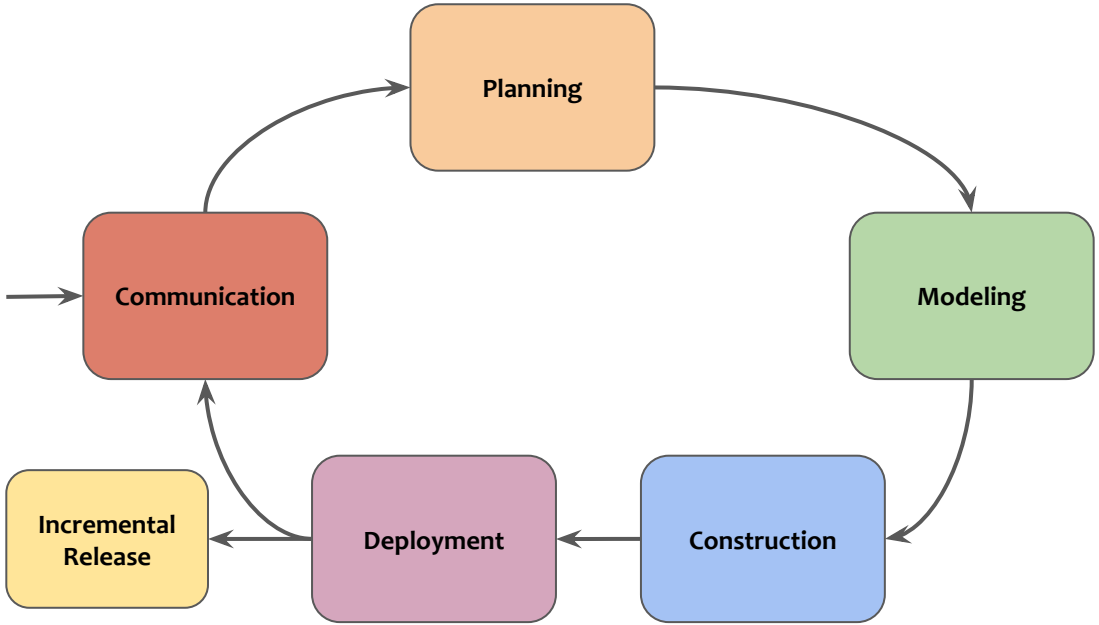
Iterative Process Flow

- Iterative process flow repeats one or more activities before proceeding to the next activity.



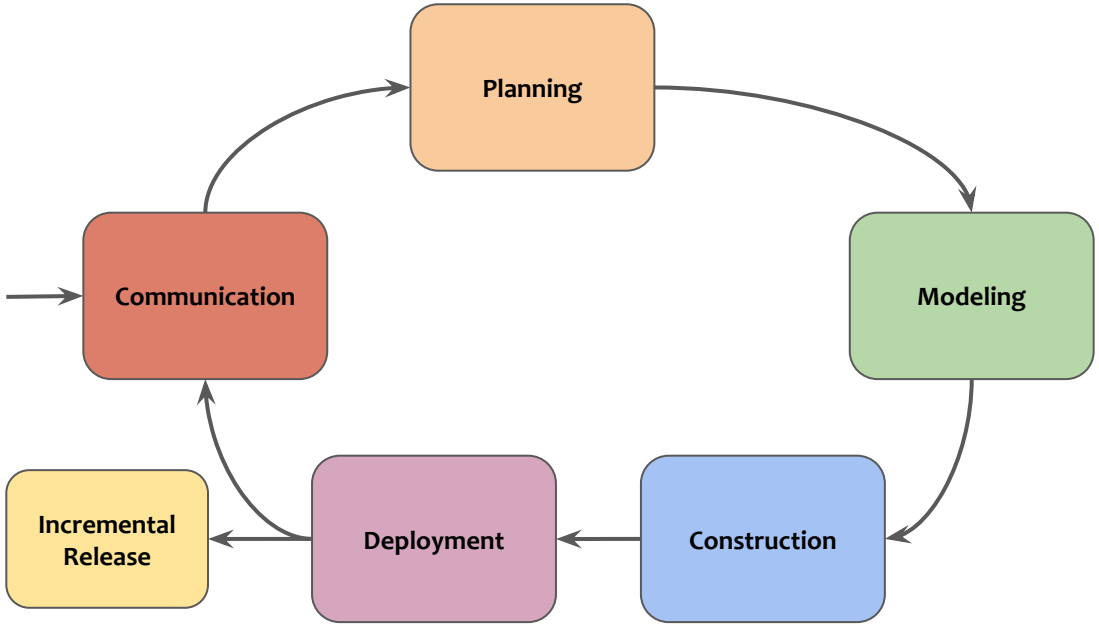
Evolutionary Process Flow

- Evolutionary process flow executes activities in a circular fashion.



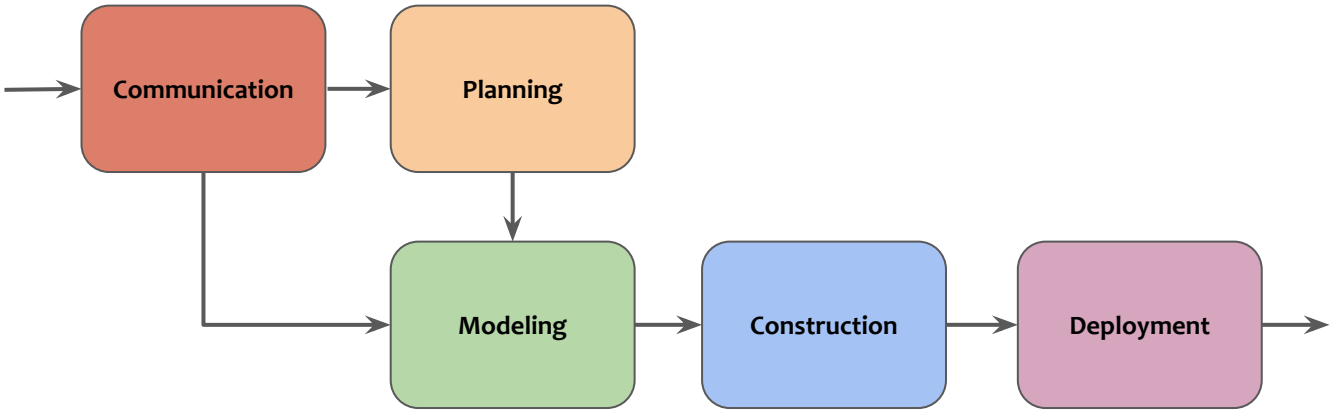
Evolutionary Process Flow

- Each cycle leads to a more complete version of the software.



Parallel Process Flow

- Parallel process flow executes one or more activities in parallel with other activities.



Prescriptive Process Models

- Prescriptive process models prescribe different workflows among the framework and other activities in a software engineering project.

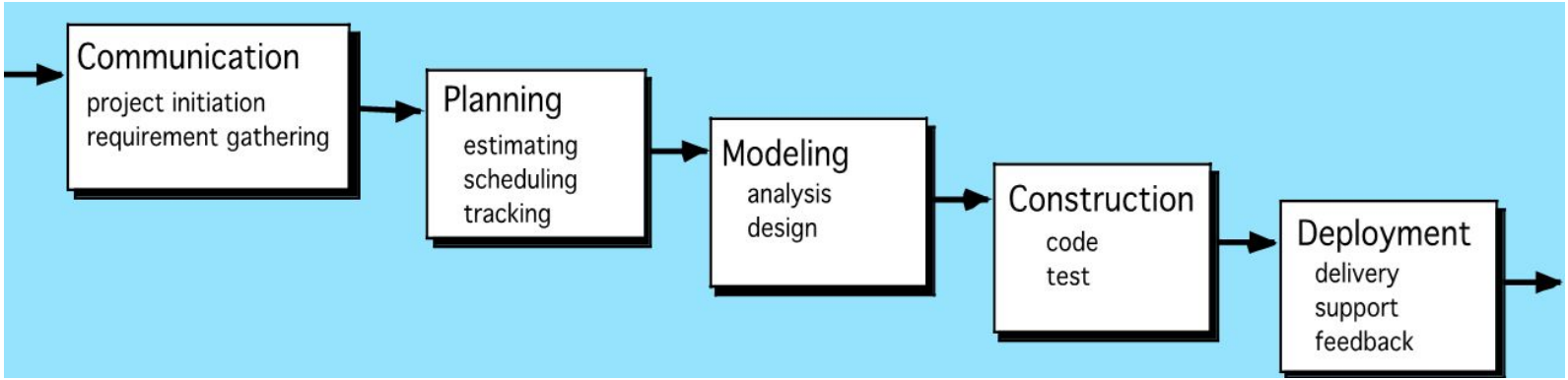
- These workflows, based on generic process models, guide a software project from its inception to its end.

Prescriptive Process Models

- In this course, we will discuss the following prescriptive process models.

    1. Waterfall Process Model
    2. Incremental Process Model
    3. Evolutionary Process Model
        a. Prototyping Model
        b. Spiral Model

Waterfall Process Model



Waterfall Process Model

- The waterfall process model is normally used in small projects in which requirements from clients are well-defined and known before the starting of the project.

- Different activities are executed in a sequential and systematic manner.

Waterfall Process Model

- The process begins with communication, where requirements are collected from the client and documented.
- In planning phase, the time and financial constraints of the project are estimated, resulting in a schedule and a budget.
- Then, a design of the software product is crafted in modeling phase based on gathered requirements and keeping project constraints in mind.

Waterfall Process Model

- In construction phase, essential code is generated and tested to build the final product.

- Finally, in deployment phase, the product is delivered to the client and necessary maintenance is provided based on the client feedback.

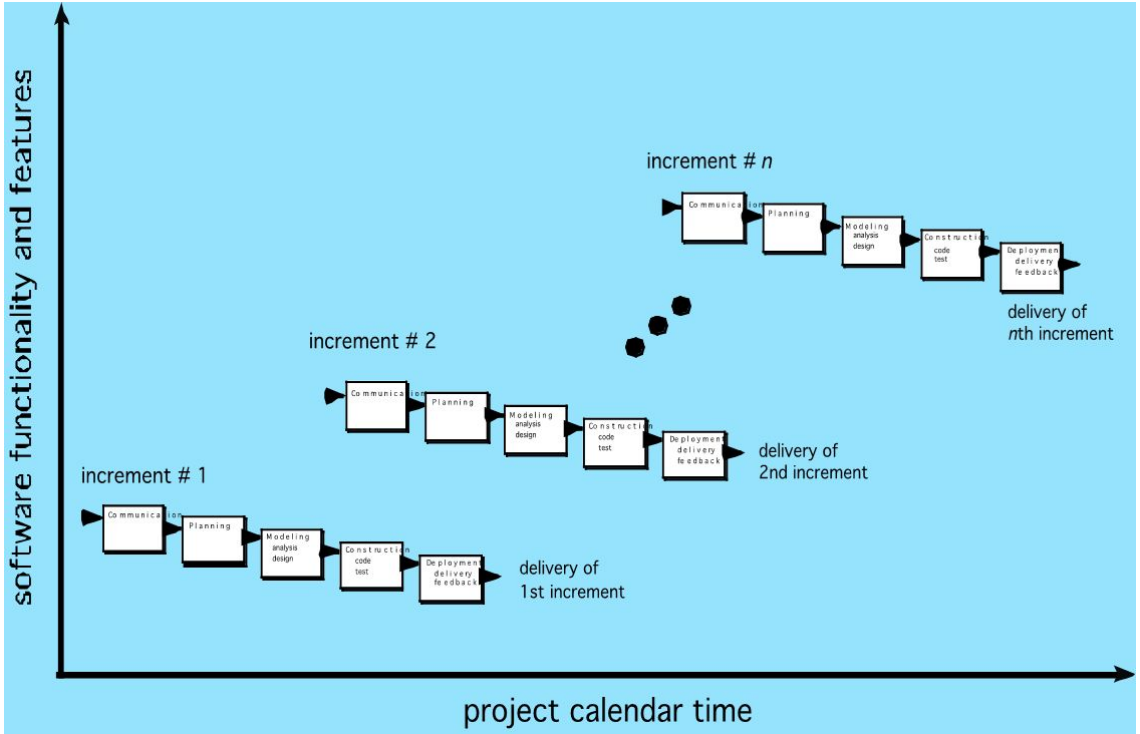
Waterfall Process Model

- Advantages of the waterfall model are as follows.
  - Simple model to use and implement.
  - Easily understandable workflow.
  - Easily manageable as requirements are known before the starting of the project.

Waterfall Process Model

- Disadvantages of the waterfall model are as follows.
    - It may become tough for the client to provide all the requirements beforehand.
    - Testing and client evaluation are carried out in the last phases resulting in high risk.
    - Iteration of activities is not promoted which may be crucial for some projects.

Incremental Process Model



Incremental Process Model

- The incremental process model involves development and deployment of a series of versions of the software product, known as increments.

- A simple functioning system, known as core product, which handles basic requirements is first developed and delivered.

- Client feedback is collected after each incremental delivery to incorporate in the next increment.

Incremental Process Model

- Multiple increments are delivered by adding more functionalities, as per the client requirements, until the final version of the product is released.

- This process model is normally used when client demands a model of the product with primitive functionalities quickly.

Incremental Process Model

- Advantages of the incremental model are as follows.
  - Flexible to changing requirements.
  - Modifications can be made throughout the process
  - Errors are mitigated as the product is assessed by the client after each incremental delivery.

Incremental Process Model

- Advantages of the incremental model are as follows.
    - Functioning software product is available at the early stage of the process.
    - Product can easily be tested because of multiple iterations.
    - The initial project cost is lower.

Incremental Process Model

- Disadvantages of the incremental model are as follows.
  - Breaking the problem into increments is difficult.
  - Total project cost is high.
  - A complete planning of the project is required before committing.
  - Refining requirements in each iteration may affect the software architecture.

Evolutionary Process Model

- The evolutionary process model is suitable when the client requirements may change and the deployment of a complete software product can not be done before the project deadline, but the delivery of a limited version of the product is possible beforehand.

- In the incremental process model, client requirements are specified beforehand and these requirements get refined in each increment.

Evolutionary Process Model

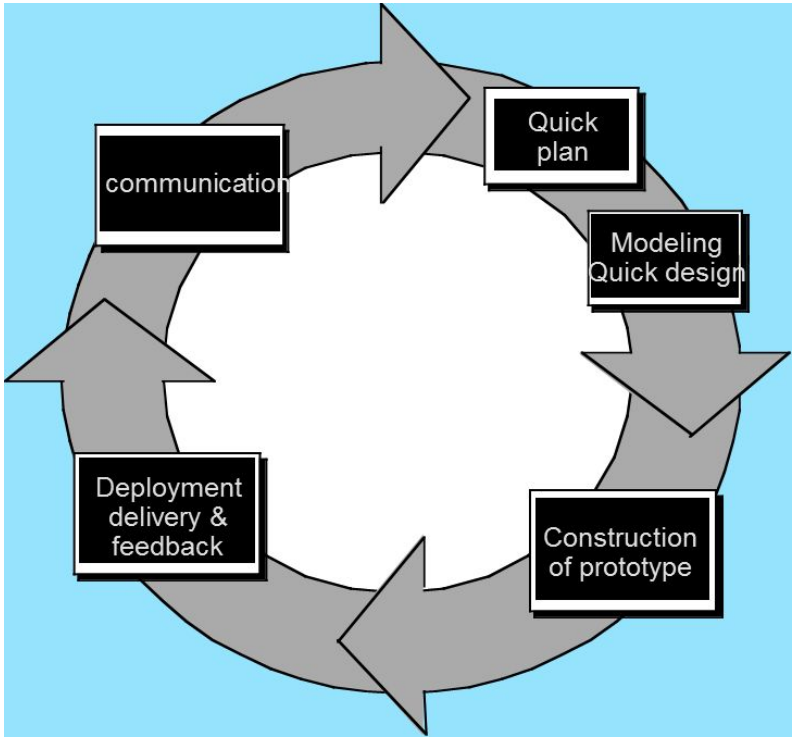
- On the other hand, the evolutionary model allows client requirements to evolve over time.

- In this course, we will discuss the following evolutionary models.

    1. Prototyping Model

    2. Spiral Model

Prototyping Model



Prototyping Model

- When the client requirements are vague and likely to change or there are confusion about working of a module of software product, a possible solution is to build a prototype of the product and then, assess that prototype.

- In the prototyping model, one or more prototypes of the software product are built with currently known client requirements before the development of final product.

Prototyping Model

- The client evaluates the prototype and provides feedback as well as additional requirements which get incorporated in next prototype.

- This workflow is repeated until the prototype evolves into a complete product, acceptable to the client.

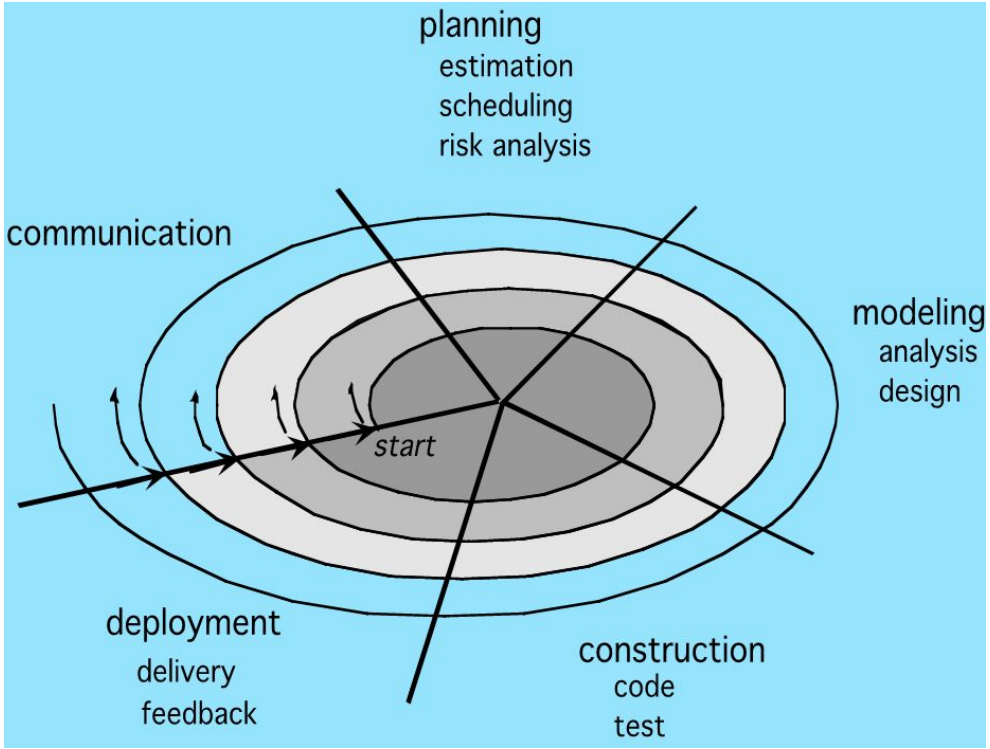
Prototyping Model

- Advantages of the prototyping model are as follows.
  - Promotes active involvement of the client.
  - Client feedback helps to better understand the product and facilitates early detection of error in the product.
  - Detailed client requirements are not needed to start the project.

Prototyping Model

- Disadvantages of the prototyping model are as follows.
  - Prototyping can slow down the process.
  - Frequent changes may increase complexity of the system.
  - Client dissatisfaction may lead to scrapping of multiple prototypes.

Spiral Model



Spiral Model

- Just like incremental process model, the spiral model refers to the development of the software product through a series of versions of that product.

- But, unlike incremental process model as well as prototyping model, this model deals with the uncertainties in software project by incorporating different risk analysis techniques throughout the process.

Spiral Model

- The first loop may result in the development of a basic prototype of the final product.
- The subsequent loops may result in the gradual development of more mature versions of the product.
- This spiral continues until an acceptable software product is built and delivered to the client.

Spiral Model

- Advantages of the spiral model are as follows.
  - Rigorously tackles risks associated with a project.
  - Any type of changes can be incorporated even at a later stage of the process.

Spiral Model

- Disadvantages of the spiral model are as follows.
  - Process gets costly and complicated.
  - Requires risk assessment expertise.

Food for Thought

- Prescriptive process models are originally introduced to bring order and structure to a software process.
- But, won't some extent of chaos and randomness beneficial to bring about creativity?

Food for Thought

- Are order and structure imposed by prescriptive models inappropriate for a software industry thriving on rapid changes?

- But, will the replacement of the traditional process models with something a bit chaotic make it impossible to coordinate the activities in a software project?

Agile Development

Why Agile?

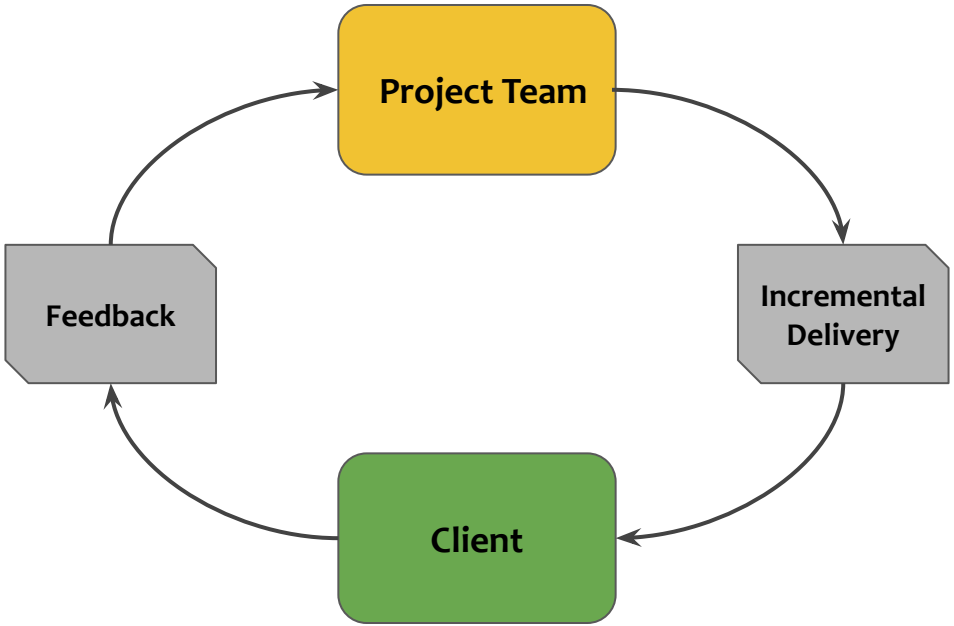
- Process models that we have discussed so far are slow, extremely slow.
- But, the modern business environment is fast-paced and ever-changing.
- So, in software development, we need a process model that is faster.

What is Agile?

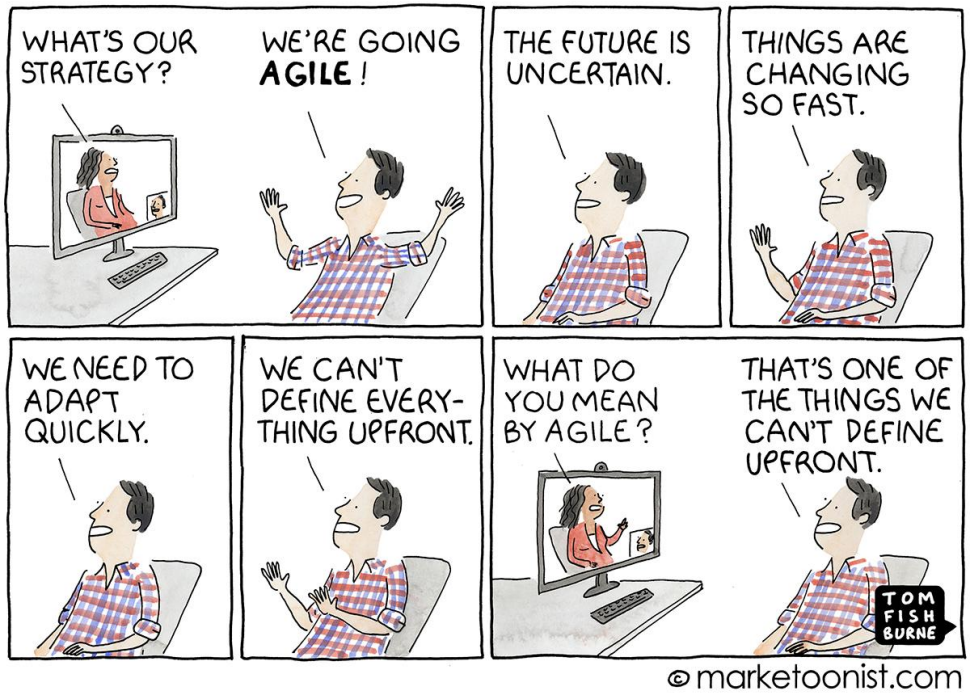
What is Agile?

- Agile means “moving quickly”.
- In Agile, the focus is on rapid and incremental delivery of software product.
- The development guideline emphasizes on delivery over analysis and design, as well as active and continuous communication between project team and client.

What is Agile?



Agile Software Development



Agile Software Development

- Agile process models put focus on quick, frequent, and incremental release of a software product.
- These process models also encourage more involvement of client in a software project.
- But, they discourage detailed analysis, planning, and documentation in a project.
- Agile software process is more flexible to changes both in project team and product.

Agile Process

- Agile process is driven by client requirements for the software product.
- These requirements tend to change quite frequently.
- The project team needs to recognize that the plans made are short-lived.
    - Some features may persist.
    - Some features may change.
    - Some features may no longer be required.

Agile Process

- This uncertainty enforces the software process to adapt to unpredictable changes.
- To do this, project team puts more focus on the construction phase than planning and modeling phases.
- Software product is made iteratively with multiple increments based on client feedback for adaptation.

12 Agile Principles

According to the **Agile Manifesto (Kent Beck *et al.*, 2001)**, there are 12 principles that guide a software project in agile manner.

1. **Our highest priority is to satisfy the customer** through early and continuous delivery of valuable software.

2. **We welcome the changing requirements,** even late in the development, to harness changes for the client's competitive advantage.

3. **We deliver working software frequently,** from a couple of weeks to a couple of months, with a preference for the shorter timescale.

12 Agile Principles

According to the **Agile Manifesto (Kent Beck *et al.*, 2001)**, there are 12 principles that guide a software project in agile manner.

4. **Client and developers must work together regularly** throughout the project.

5. **We conduct projects around motivated individuals** by giving them the environment and support they need, and trusting them to get the job done.

6. The most efficient and effective method of conveying information to and within a project team is **face-to-face conversation.**

12 Agile Principles

According to the **Agile Manifesto (Kent Beck *et al.*, 2001)**, there are 12 principles that guide a software project in agile manner.

7. **Working software** is the primary measure of progress.

8. Agile process promotes sustainable development through maintaining **a constant pace indefinitely.**

9. **Continuous attention to technical excellence and good design** enhance agility.

12 Agile Principles

According to the **Agile Manifesto (Kent Beck *et al.*, 2001)**, there are 12 principles that guide a software project in agile manner.

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 project teams.**

12. At regular intervals, the project team reflects on how to become more effective, **then tunes and adjusts its behavior accordingly.**

Human Factors in Agile Process

- The software process reshapes according to the need of people and teams involved in a project, not the other way around.
- But, in order to lead a software process to a success, people involved must grow some traits.
- In agile process, people need to be competent, skillful, knowledgeable, committed, collaborative, decisive, faithful, respectful, and self-organized.

Agile Process Models

- In this course, we will discuss the following agile process models.

  1. Extreme Programming (XP)
  2. Scrum
  3. Kanban (看板)

Extreme Programming (XP)

- Extreme programming (XP), developed by Kent Beck, gets its name from the concept that whatever the beneficial software engineering practices are, they should be taken to *extreme* levels, from day one.

- XP is lightweight because it is based on the idea of exploring *the simplest thing that will work* without putting too much weights on the long-term view of software product.

- This process model values communication with clients, simplicity of project, feedbacks from the client, and mutual respect between project team and client.

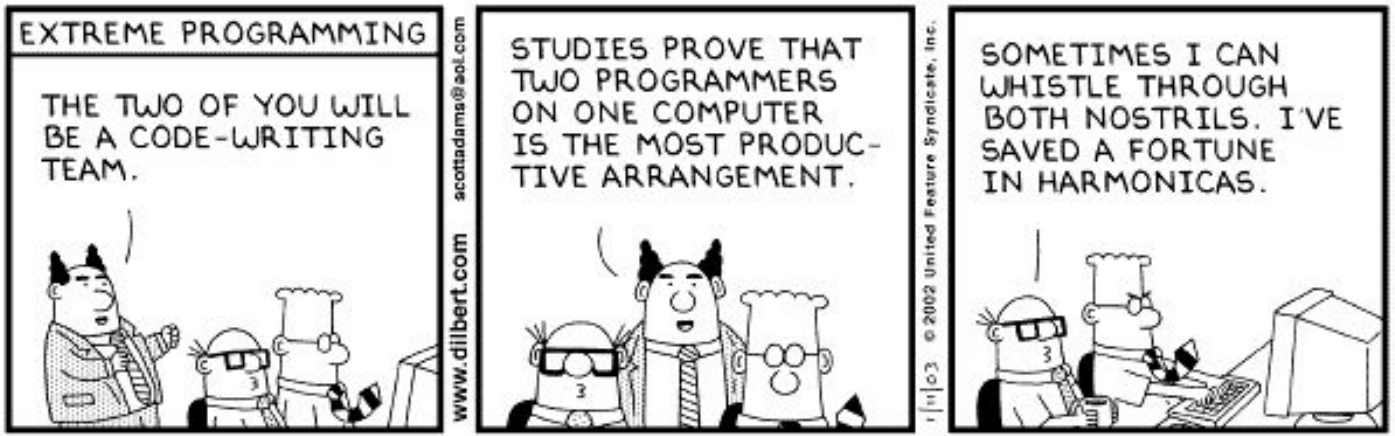
Extreme Programming (XP)

- And, this model prioritizes customer satisfaction over everything else.

- Teamwork is extremely important in XP since the problems, associated with a project, are solved by the whole team.

- XP promotes *pair programming* in which two software developers team together on one computer and work together to design, code as well as test client requirements, also known as *user stories*.

Extreme Programming (XP)



Extreme Programming (XP)

The advantages of XP are as follows.

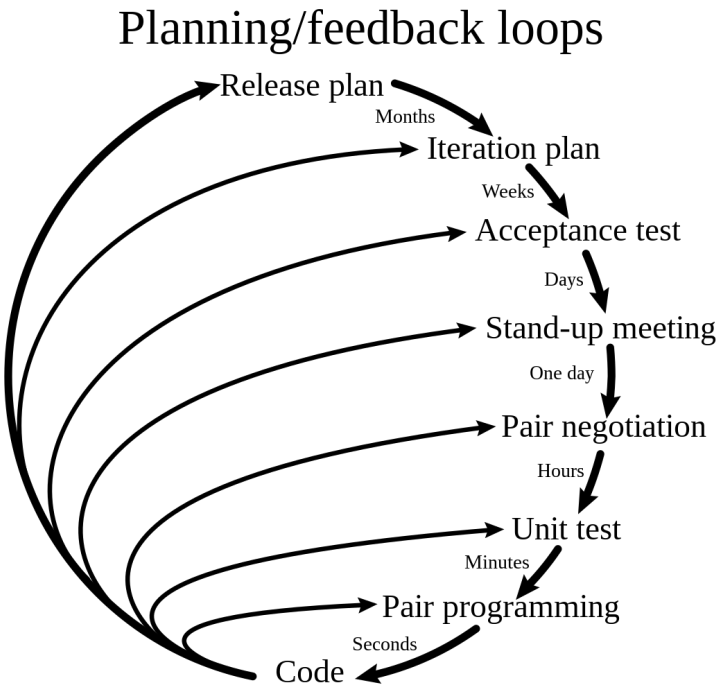
- XP encourages simple code which allows modification at any given time.
- This process model promotes testing codes from day one, resulting in more agile software development.
- XP maintains an energizing and uplifting environment for developers within a project team.

Extreme Programming (XP)

The disadvantages of XP are as follows.

- The extreme focus on coding can lead to neglecting design, resulting in degradation of software product quality.
- Lack of documentation and monitoring may lead to repetition of similar error in the future.

Extreme Programming (XP)



Source: Wikipedia

Scrum

Scrum

- Scrum is a position in rugby in which players from two competing teams pack together closely to gain control of the rugby ball.
- In project management, however, scrum is a lightweight, iterative, and incremental agile process model.
- That means, scrum prioritizes simplicity in a project and develops product in a gradual manner with frequent delivery.
- Currently, scrum is one of the widely used process models in software industry.

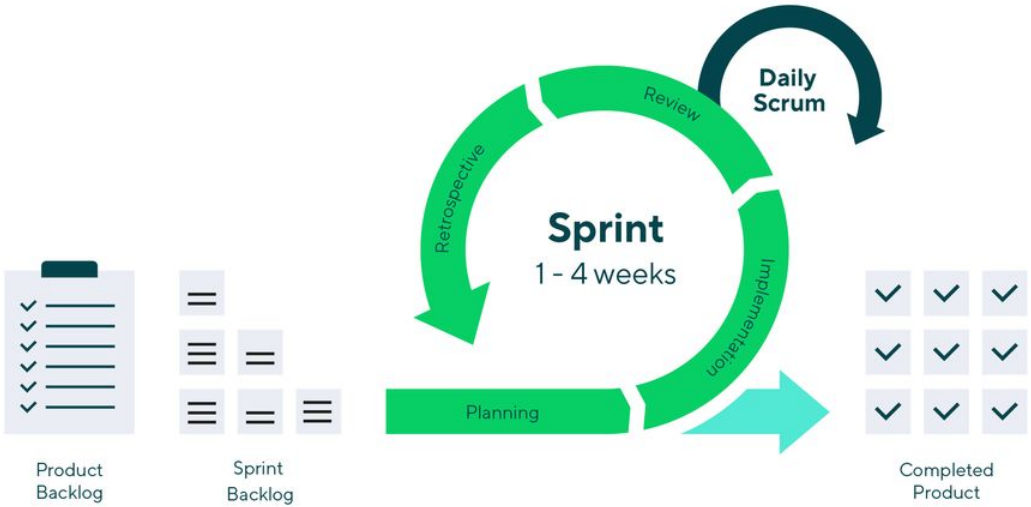
Scrum

- There are some keywords associated with scrum.
    - **Sprint** is the unitary cycle or iteration of project development phases. It usually lasts from 2 weeks to 1 month during which participants work on certain aspects of a project.
    - **Backlog** means the requirements from client for the product. There are 2 (two) types of backlog including product backlog and sprint backlog. **Product backlog** means the overall client requirements in a project. On the other hand, **Sprint backlog** means the client requirements that will be addressed in a particular sprint.

Scrum

- There are some keywords associated with scrum.
    - **Daily scrum** means a meeting conducted, on a daily basis, within a project team for synchronization. In it, plans and progress of the project are discussed.
    - **Scrum master** is usually the project manager and **Product owner** basically means the client.
    - **Tickets** are subtasks in a project and **Scrum board** is maintained to keep track of them.

Scrum



Source: https://www.wrike.com/scrum-guide/scrum-sprints/

Scrum

- A particular sprint, in scrum, starts by analyzing the product backlog and deciding on which requirements to implement in the current sprint. Then, the selected requirements are added to the sprint backlog.

- After that, the project team starts working on the sprint backlog in 4 (four) phases including planning, implementation, product review, and process review (aka retrospective).

- Finally, the completed portion of the product is delivered to the client for feedback. The unfinished sprint backlogs are sent back to the product backlog for the next sprint.

Scrum

- The advantages of scrum are as follows.
    - Scrum encourages less authority and hierarchy in a project team, thus, giving developers more freedom to express themselves and work.
    - This process model focuses on adaptability to tackle rapid changes in client requirements.
    - In scrum, the products usually take less time to become deliverable.

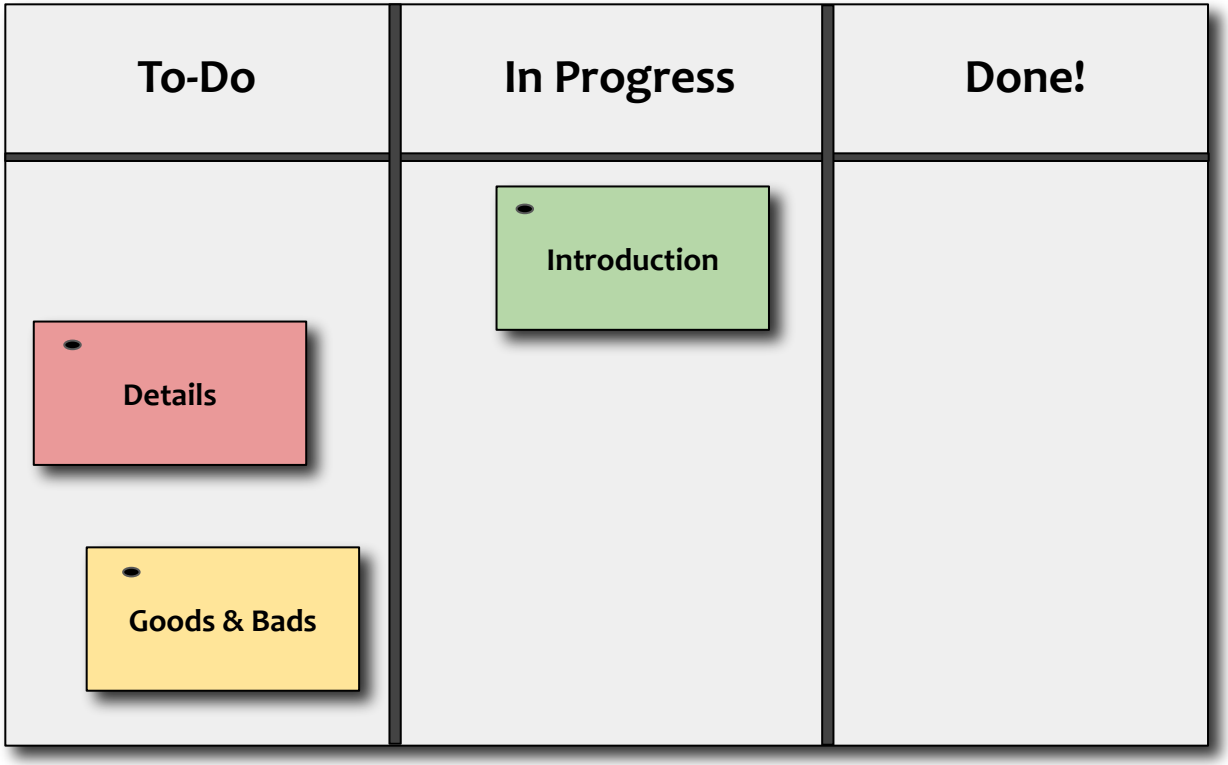
Scrum

- The advantages of scrum are as follows.
    - Scrum enforces more client involvement in a project.
    - In scrum, different aspects of current sprint is reviewed in retrospective phase, thus, ensuring less mistakes in next sprint.

Scrum

- The disadvantage of scrum is as follows.
    - Scrum usually works well with small project team.

Kanban



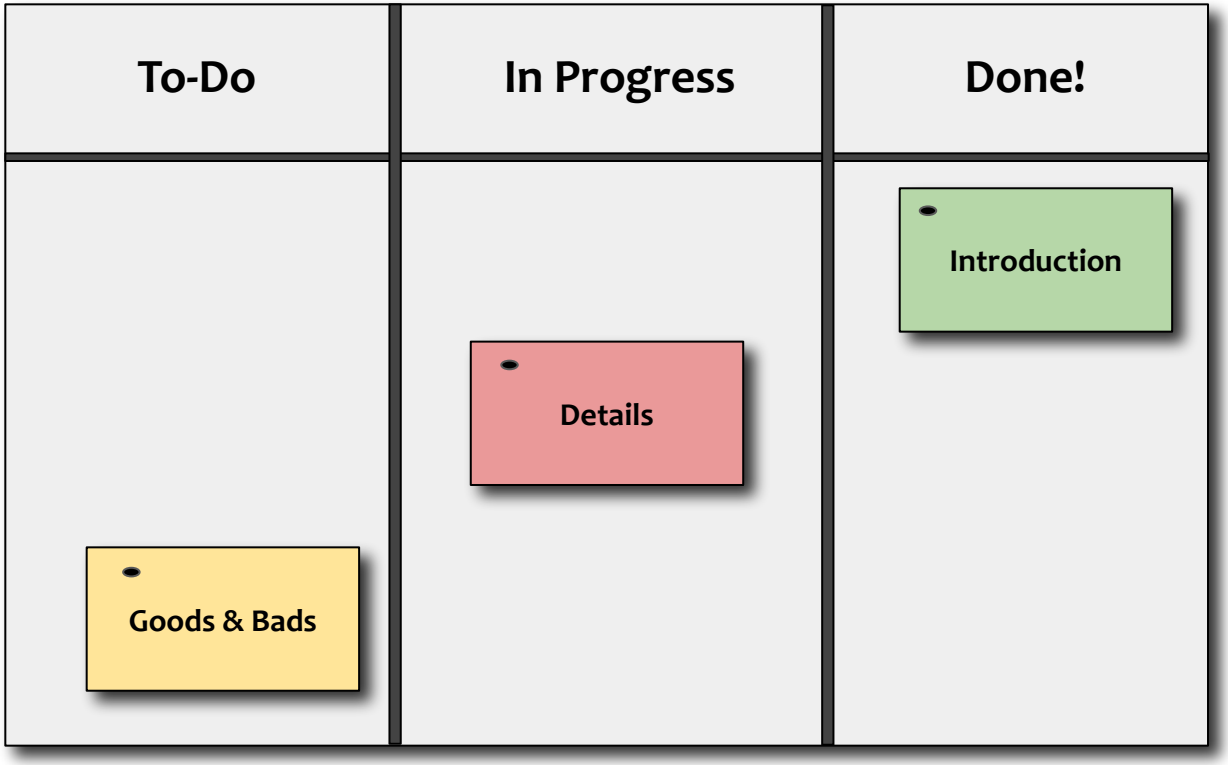
Kanban

Source: https://abandonedkansai.com/2017/02/14/wakayama-mountain-hotel/handwritten-japanese-drink-menu/

Kanban

- In Japanese, the word Kanban (看板) means signboard which is visible to everyone around it.
- In project management, however, Kanban is another agile process model.
- Kanban puts heavy focus on the visualization of different aspects of a project so that everyone involved in a project is kept up-to-date all the time.

Kanban



Kanban

- In kanban, just like scrum, different tasks in a project are further divided into modular subtasks.
- Each of such subtasks is, then, assigned to a **Kanban card** which contains details on that particular subtask as well as associated user story (aka client requirement).
- A **Kanban board** is maintained to keep track of kanban cards.

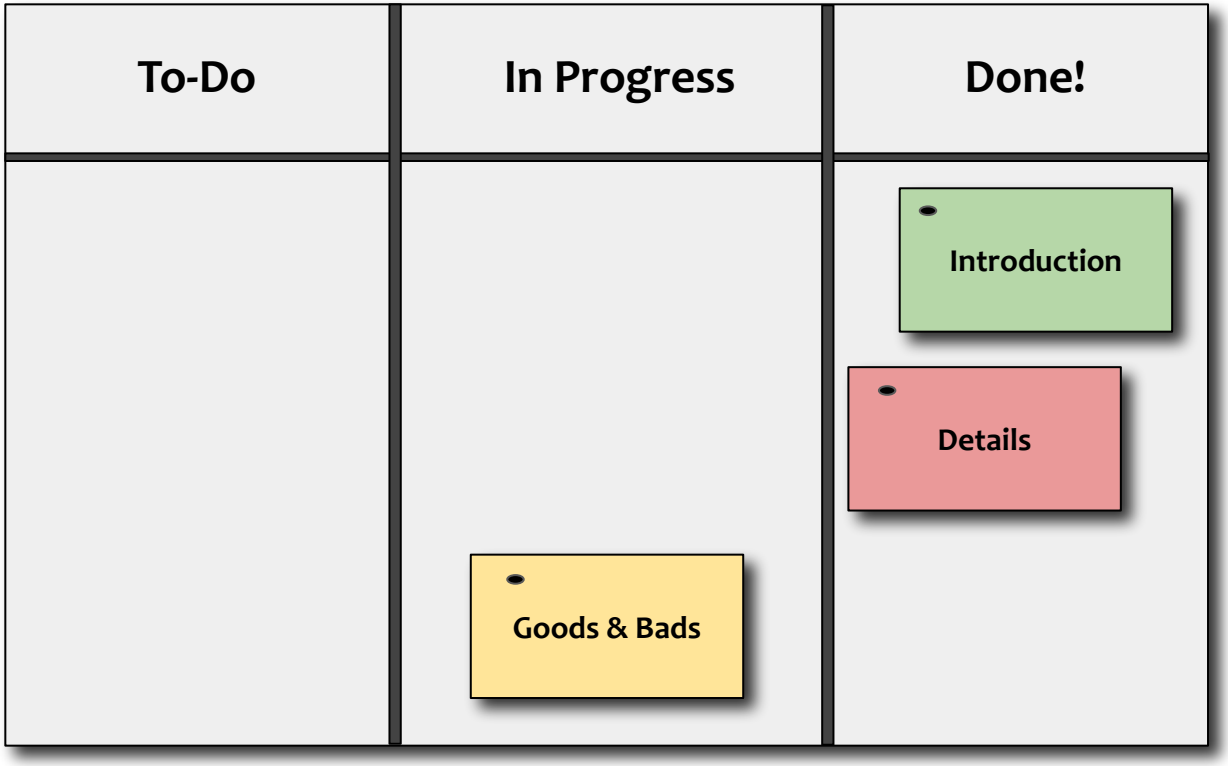
Kanban

- Like scrum, kanban follows *pull system*, that is, whenever a participant is available, he/she attempts to pull a kanban card from left to right by working on the subtask assigned.

- Similar to scrum, there is an **Agile coach** in a project team in kanban and the team conducts **Daily stand-up meeting** and retrospectives during a project.

- But, unlike scrum, kanban is a continuous process, that is, there is no such thing as sprint or iteration.

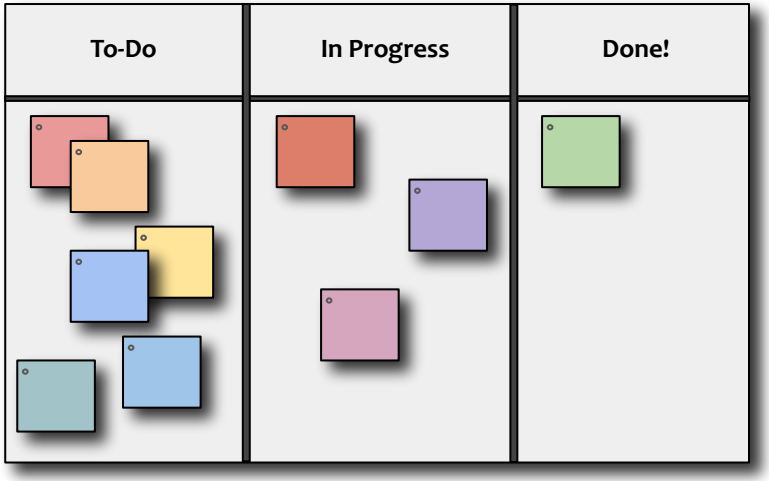
Kanban



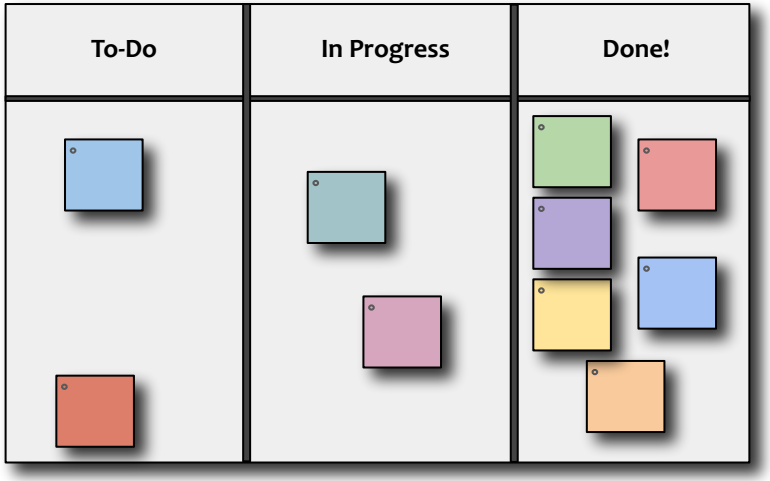
Kanban

- The advantages of kanban are as follows.
    - Kanban encourages leadership from every participant in a project, thus, creating a free environment in which everyone can express his/her opinion about project and take decisions rapidly in the process.
    - This process model can adapt to different natures of a project team.
    - Developers can adjust workload per kanban card by analyzing the kanban board to maintain a decent lead time in workflow.

Kanban



You probably should reduce the workload per kanban card to speed up the process :(

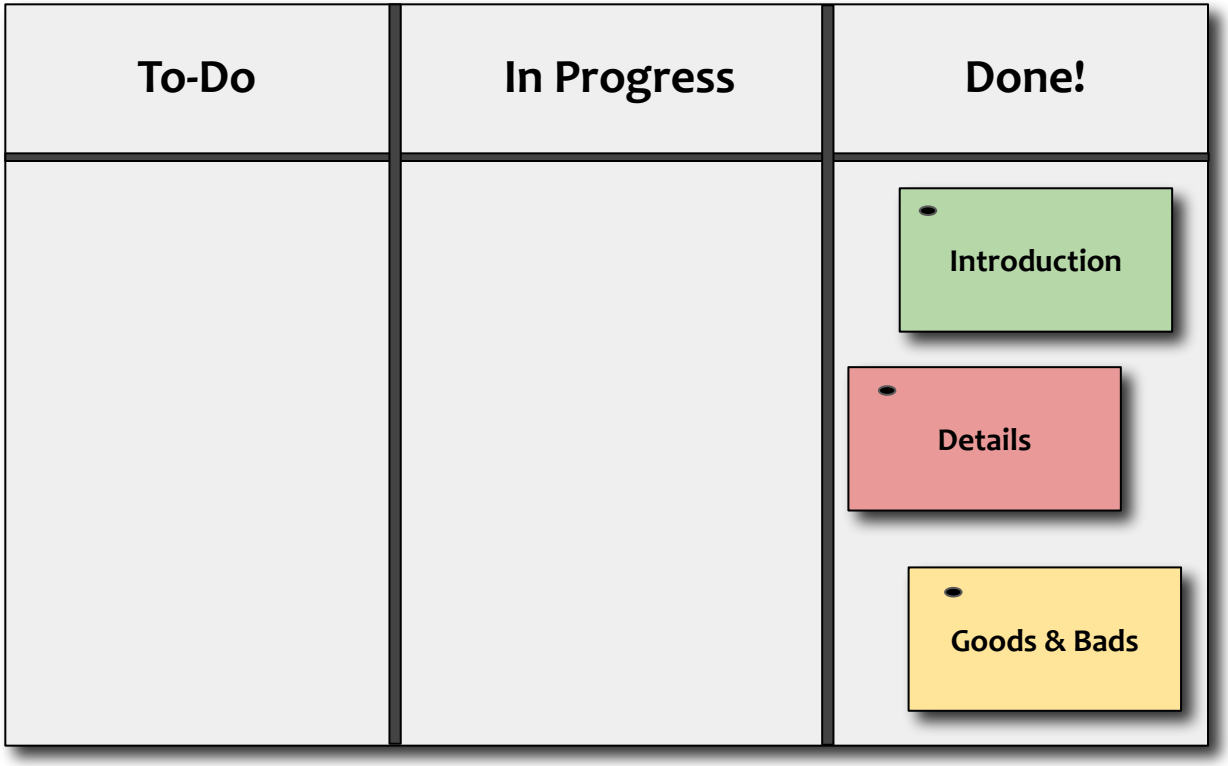


You can now increase the workload per kanban card to finish the process :)

Kanban

- The disadvantage of kanban is as follows.
    - In kanban, there is no particular timeframe which may result in delay of the project.

Kanban



Jira

- Jira, developed by Atlassian, is a widely used issue tracking product in agile project development.