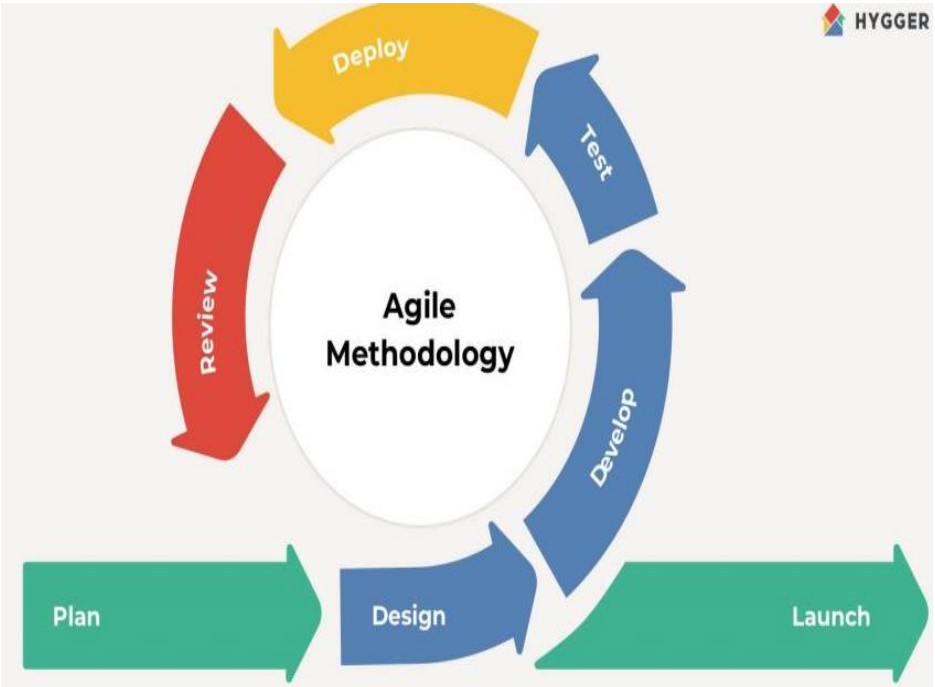
CSE 6324

Advanced Topic in Software Engineering

Farnaz Farahanipad

Agile Model

- **Agile process model**, refers to a software development approach based on an incremental, iterative approach.
- Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning.
- Agile methodologies are open to changing requirements over time and encourages constant feedback from the end users.
- The goal of each iteration is to produce a working product.
- Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.



The Principles of Agile Methods

- Iterative design process
- Continuous stakeholder engagement
- Aims for quality and reliable software
- Short development cycles(up to a month)
- Simplicity

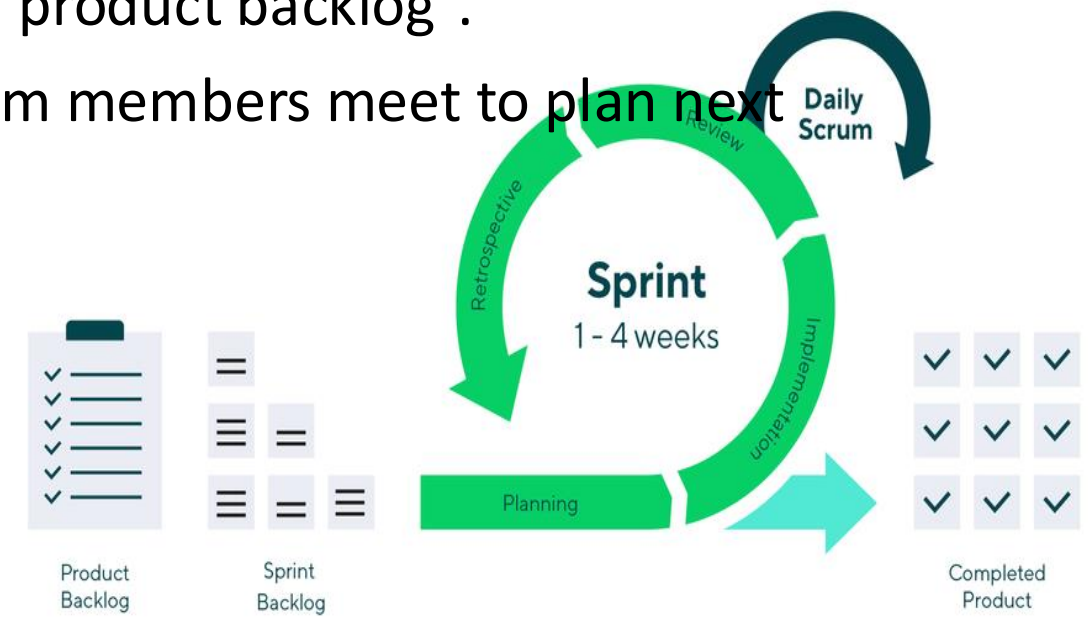
Agile Methods

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

1-Scrum Process

1-Scrum

- It is an iterative software development model used to **manage complex software and product development**.
- Fixed-length iterations, called **sprints** lasting **one to four weeks** long.
- Scrum teams do a little of everything all the time(requirement, design, code, test)
- Requirements are captured as items in a list of “product backlog”.
- At the end of each sprint, stakeholders and team members meet to plan next steps.



1-Scrum Framework

Roles:

- Product owner
- Scrum master
- Team

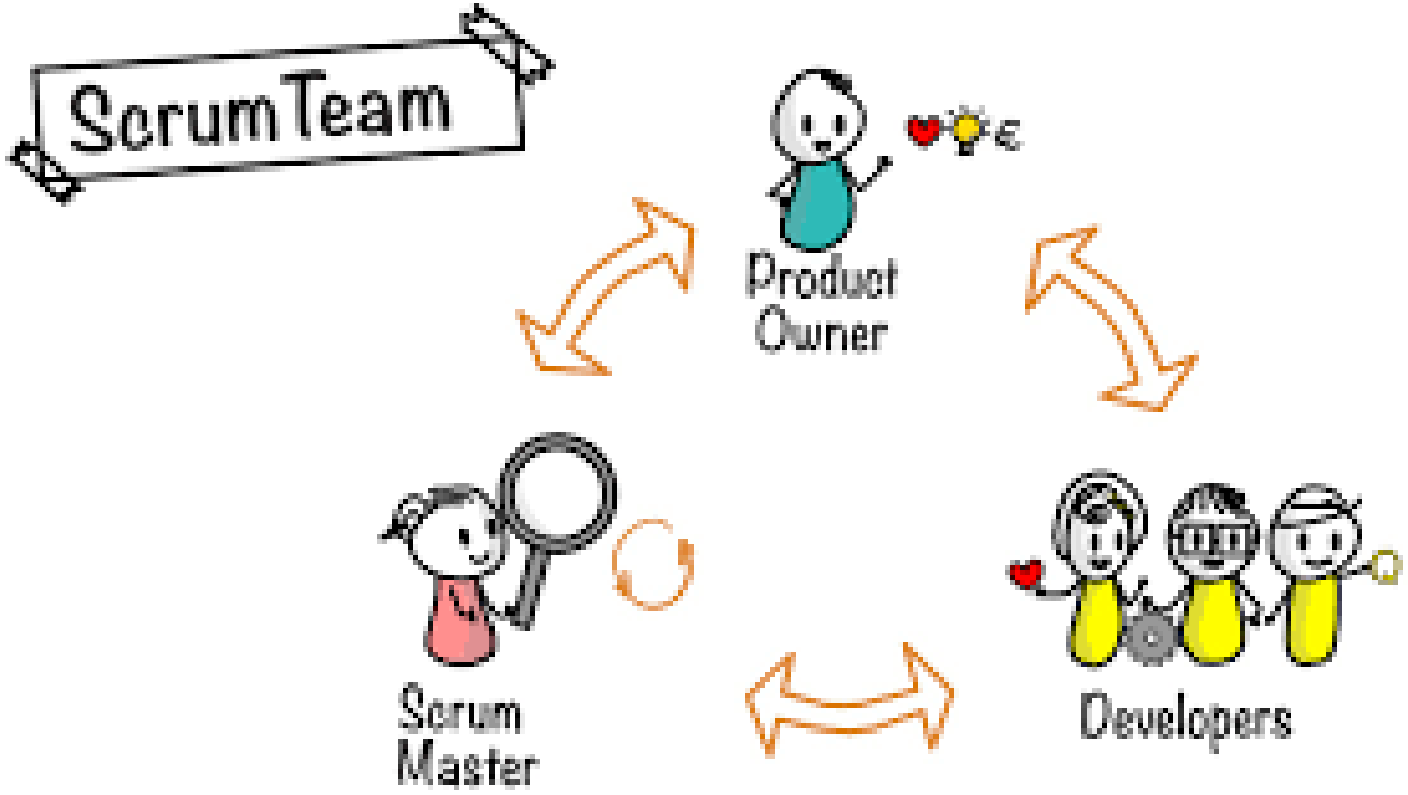
Artifacts:

- Product backlog
- Sprint backlog
- Burndown charts

Events:

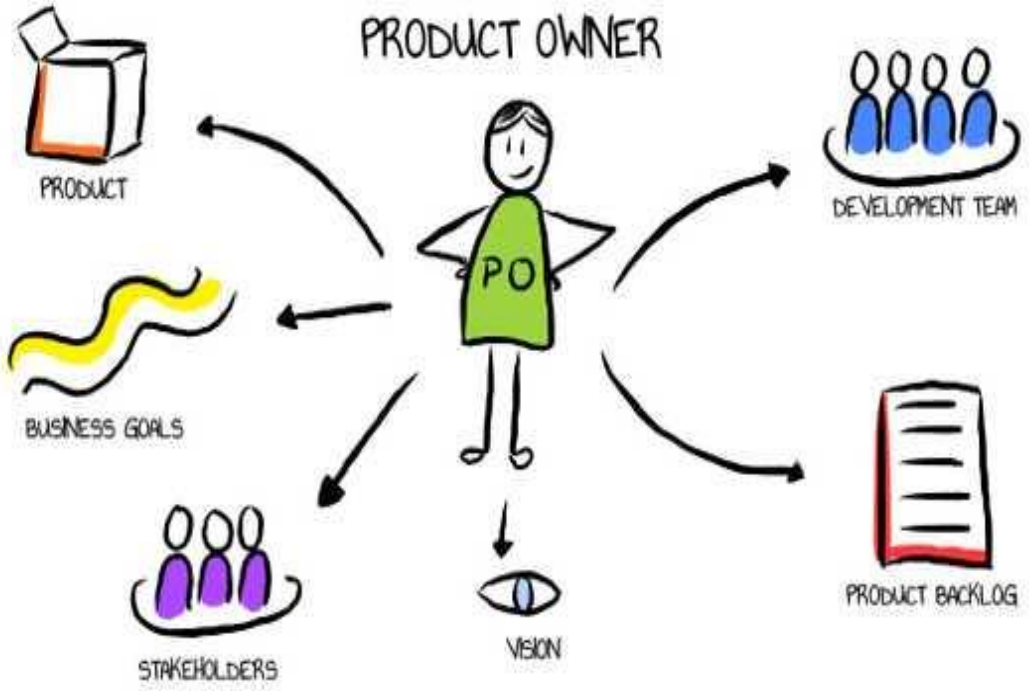
- Sprint planning
- Daily scrum meeting
- Sprint review
- Sprint retro

1-Scrum: Roles



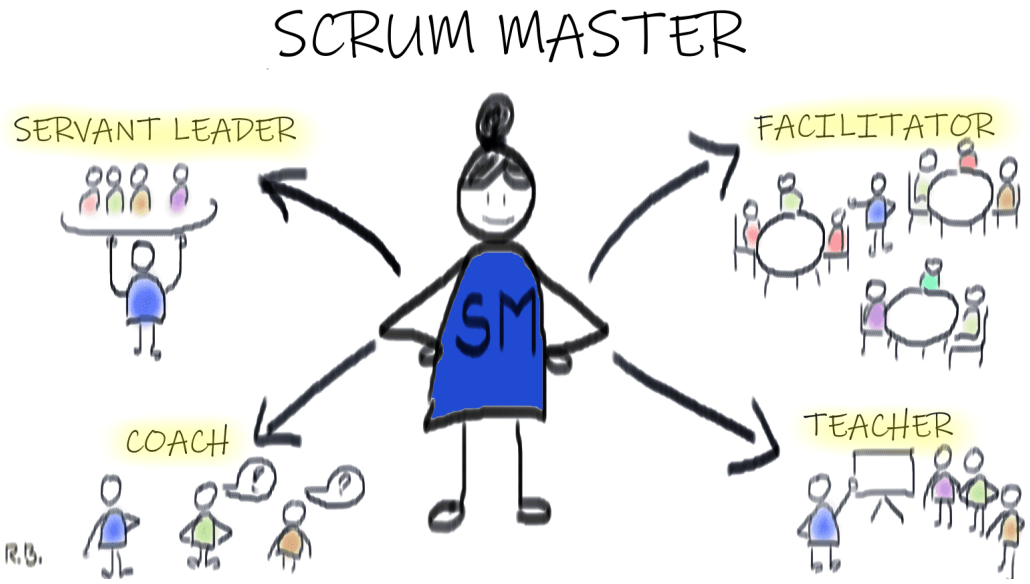
1-Scrum: Roles

- Product Owner**(PO):**
  - Define the features of the product
  - Make scope and schedule, prioritize the product backlog
  - Sharing progress during meetings
  - Accept or reject work results
  - PO represent the stakeholder and act as the voice of the customer.



1-Scrum: Roles

- Scrum Master;
  - Being a coach to the team, focusing on the Scrum principles and values;
  - Assisting the product owner in the maintenance of the product backlog and making sure that everyone understands what they need to do;
  - Guiding the team to ensure they avoid or overcome any obstacle;
  - Organizing team events to ensure progress;

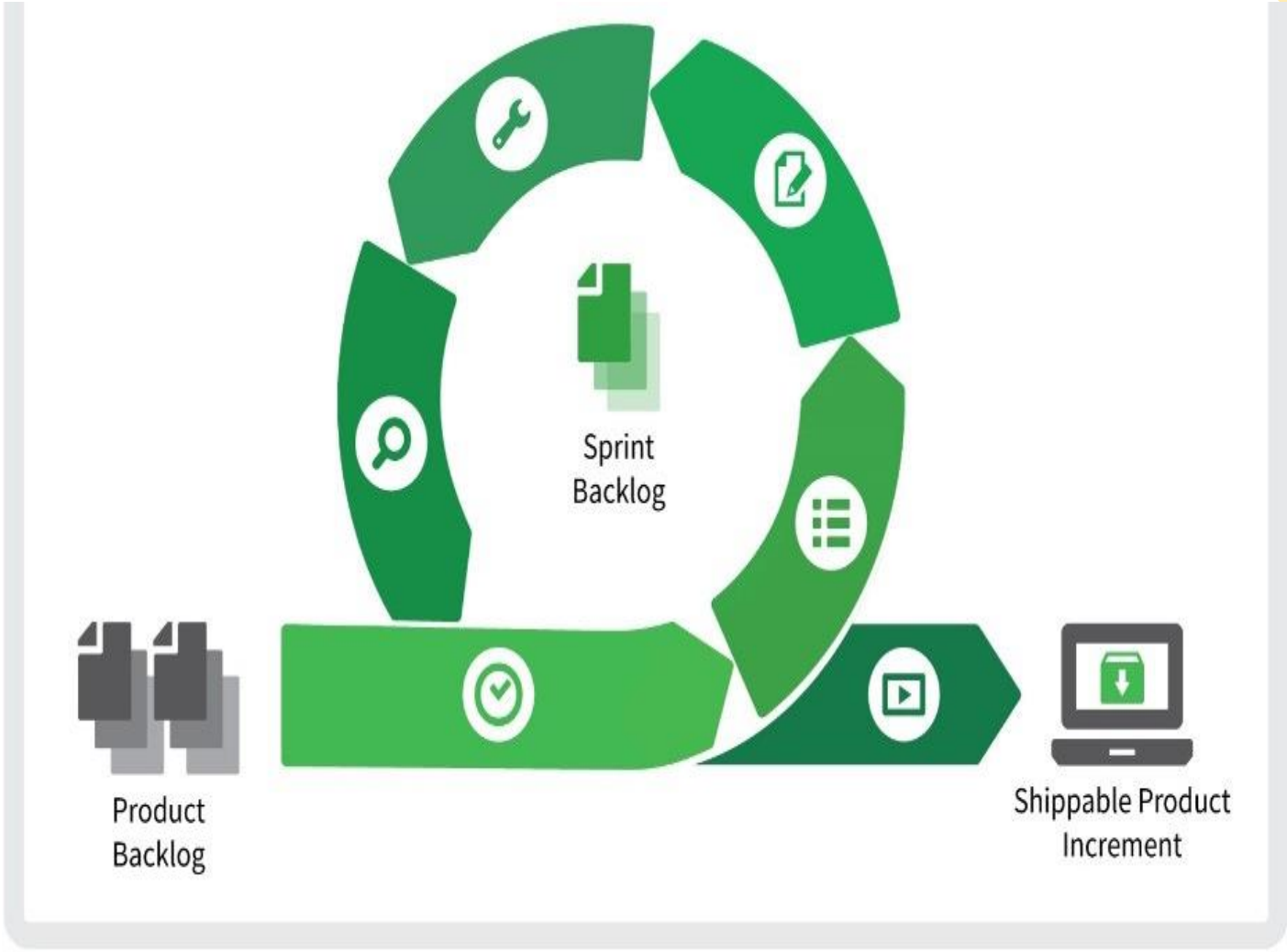


1-Scrum: Roles

- Development Team
  - 3 to 9 member
  - various skillsets
  - Self-governing

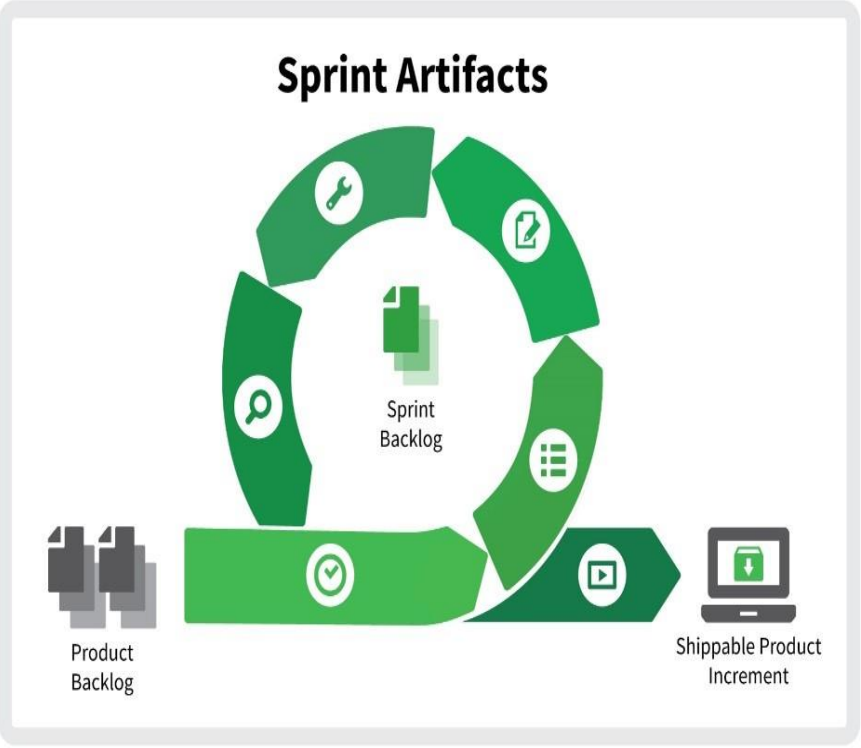
1-Scrum Artifacts

- What will the Scrum team **work on next sprint**?
- What will the Scrum **team work on this sprint** and how will they get it done?
- What will the Scrum team have made by the **end of this sprint** and how will they know it's "done"?



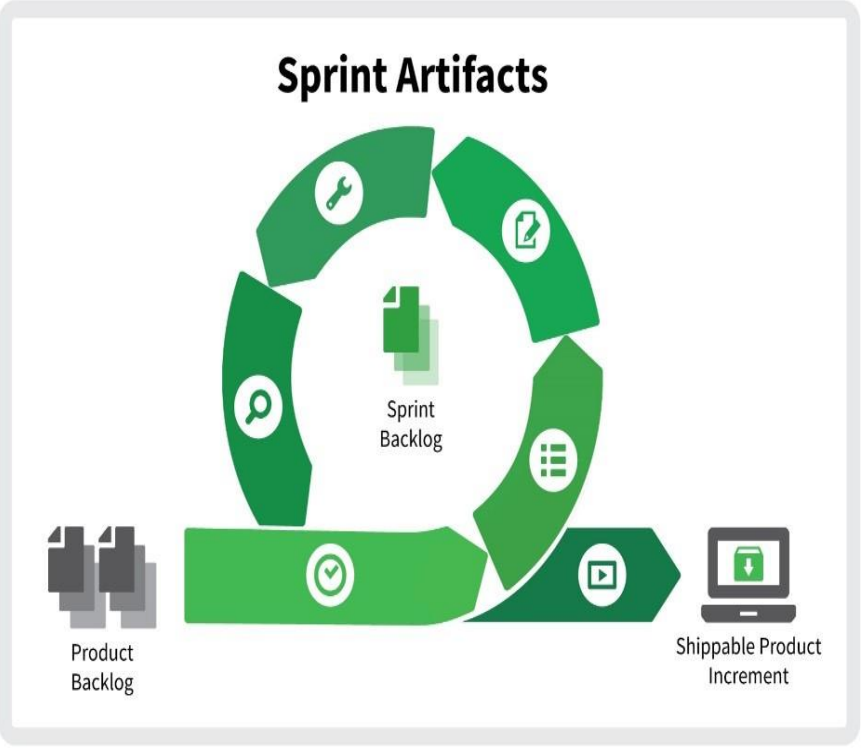
1-Scrum Artifacts

- **Product backlog**
  - ordered master to-do list for a project.
  - At the beginning of each sprint, the Scrum team plans and executes the next highest priority item on the product backlog
  - It is **a dynamic artifact** and changes as the Scrum team gains additional knowledge regarding the requirements and solution they are developing.
  - The Product Owner is responsible for maintaining the product backlog.



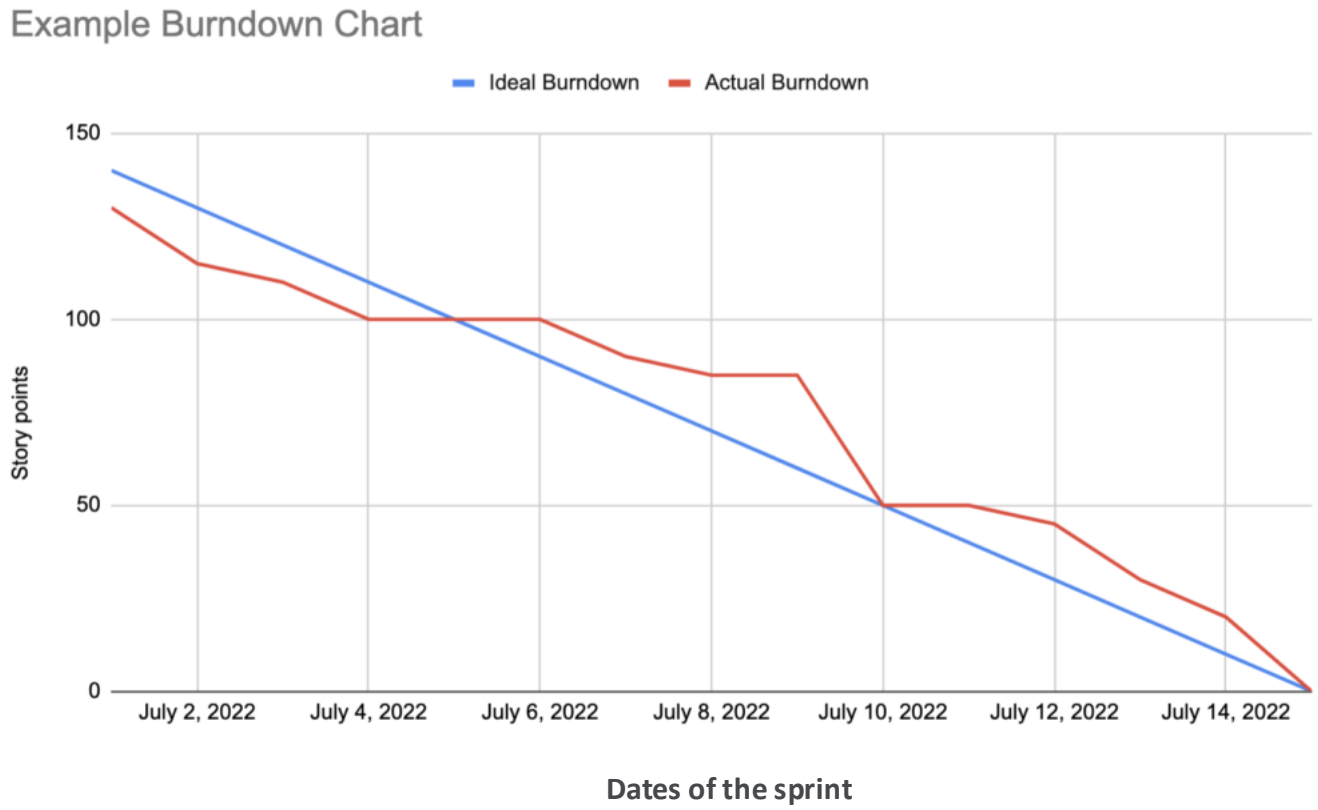
1-Scrum Artifacts

- **Sprint backlog**
  - created at the beginning of each sprint
  - includes a sprint goal and the sprint end date, a prioritized list of requirements, estimated effort for each item and a **burndown** chart displaying the status of work being performed throughout the sprint
  - The development team is responsible for managing and maintaining the sprint backlog.



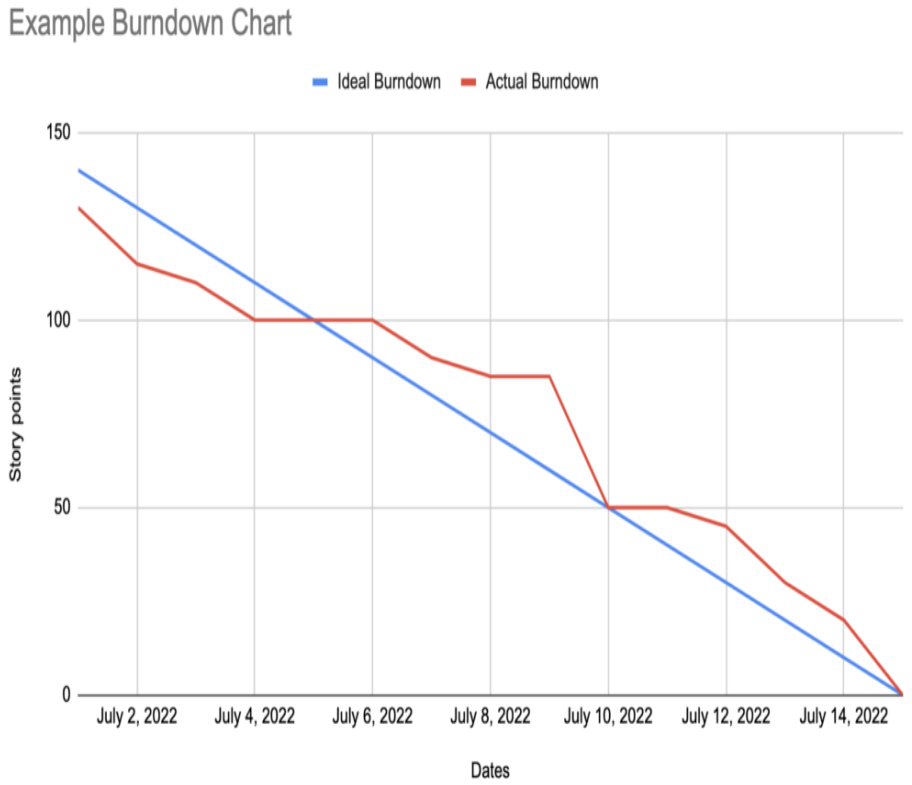
Burndown Chart*

- To track progress throughout the sprint with "work remaining" on the Y axis, and "time" on the X axis



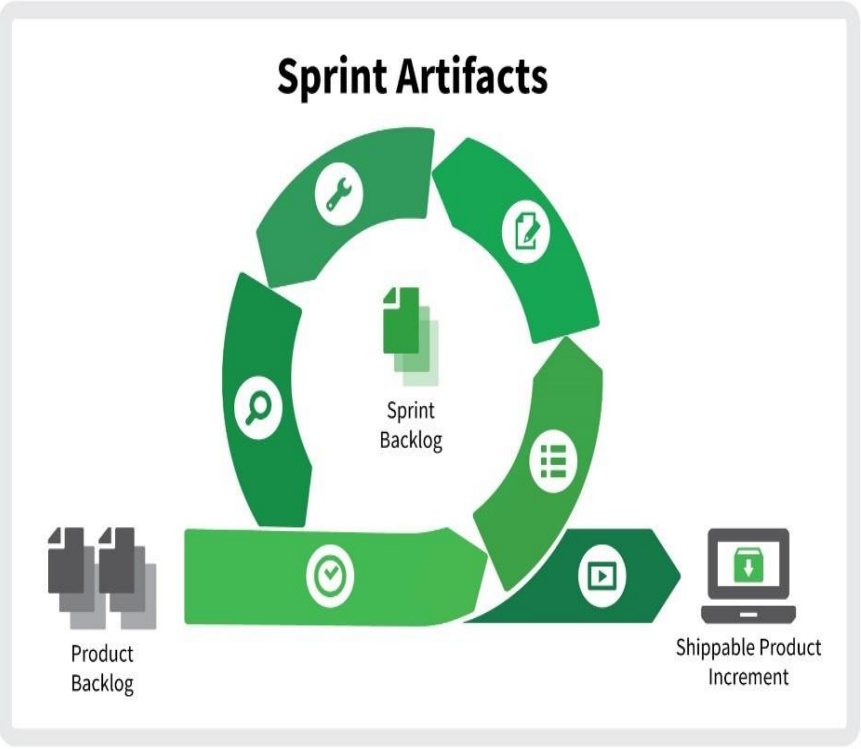
Burndown Chart*

	A	B	C
1	**Dates**	**Ideal Burndown**	**Actual Burndown**
2	July 1, 2022	140	130
3	July 2, 2022	130	115
4	July 3, 2022	120	110
5	July 4, 2022	110	100
6	July 5, 2022	100	100
7	July 6, 2022	90	100
8	July 7, 2022	80	90
9	July 8, 2022	70	85
10	July 9, 2022	60	85
11	July 10, 2022	50	50
12	July 11, 2022	40	50
13	July 12, 2022	30	45
14	July 13, 2022	20	30
15	July 14, 2022	10	20
16	July 15, 2022	0	0
17			



1-Scrum Artifacts

- **Shippable product increment**:
  - At the end of every sprint, the development team delivers working, tested and integrated functionality to stakeholders and customers to review.



1-Scrum Events

- Agile development cycles are iterative, and are often referred to as ‘iterations’. A sprint is a time-boxed iteration. Each sprint contains four key events:
    - Sprint Planning
    - Daily Scrum
    - Sprint Review
    - Sprint Retrospective

1-Scrum Events

- Agile development cycles are iterative, and are often referred to as ‘iterations’.
- A sprint is a time-boxed iteration. each Sprint contains four key events:
  - **Sprint Planning**
    - Sprint planning be held at the start of each sprint.
    - At a sprint planning meeting, the team decides what to work on next and how they will do it.
    - The product owner presents a sprint goal and the development team identifies the specific product backlog requirements that can be addressed during the sprint.
    - The Scrum master facilitates sprint planning, however, the product owner is responsible for the sprint goal and provides clarification to the team regarding requirements and business goals and objectives.
  - Daily Scrum
  - Sprint Review
  - Sprint Retrospective

1-Scrum Events

- Agile development cycles are iterative, and are often referred to as ‘iterations’.
- A sprint is a time-boxed iteration. each Sprint contains four key events:
  - Sprint Planning
  - **Daily Scrum**
    - The daily scrum or standup should be scheduled for the same time each day and be relatively brief.
    - Depending on the scale of the project a 15 or 30 minute duration can suffice.
    - Each developer states what work he or she completed the prior day so that each team member is clear on progress.
  - Sprint Review
  - Sprint Retrospective

1-Scrum Events

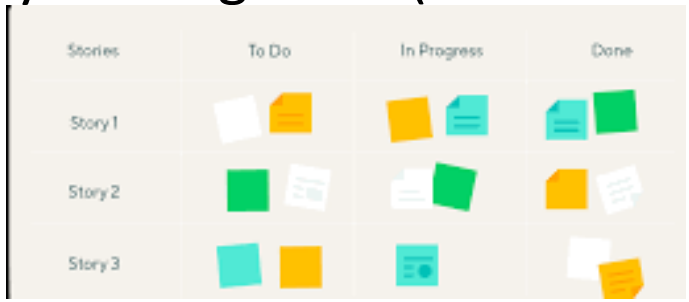
- Agile development cycles are iterative, and are often referred to as ‘iterations’.
- A sprint is a time-boxed iteration. each Sprint contains four key events:
  - **Sprint Planning**
  - Daily Scrum
  - **Sprint Review**
    - Each sprint closes with a sprint review. This allows the product owner to verify that the Scrum team is developing what the customer wants.
    - The product owner presents the sprint goal and status to the stakeholders.
    - The product owner then updates the product backlog, based on this feedback.
  - Sprint Retrospective

1-Scrum Events

- Agile development cycles are iterative and are often referred to as ‘iterations".
- A sprint is a time-boxed iteration. each Sprint contains four key events:
  - **Sprint Planning**
  - Daily Scrum
  - Sprint Review
  - **Sprint Retrospective (Retro)**
    - Agile teams hold retrospective meetings after a time-boxed period of work is complete
    - provides the Scrum team with the opportunity to **inspect** and **adapt** its **processes**, environment, tools, communications, and obstacles in order to make improvements.
    - Action plans for adopting those changes are developed and added to the product backlog to be implemented in a future sprint.
    - The overall goal of the retrospective is to fix any problems in the team.

1-Tools in Scrum

- **Scrum board**: The Scrum board helps to visualize your sprint backlog and traditionally involves index cards or Post-It notes on a whiteboard. The board is usually divided into three categories: to do, work in progress, and done. The team updates the board by moving tasks (written on cards) through the columns on the board.



- **User stories**: represent the smallest unit of work in the framework.
- **Timeboxing**: A timebox is a set period of time that a team works towards completing a goal. Instead of letting a team work until the goal is reached, the timebox approach stops work when the time limit is reached.

1- Advantages and Disadvantages of Scrum

?

1- Advantages and Disadvantages of Scrum

Advantages	Disadvantages
Unstable requirements do not hold up progress.	Adopting the Scrum framework in large teams is challenging
The whole team have visibility of everything	The framework can be successful only with experienced team members
Customers see on-time delivery of increments and gain feedback on how the product works	Daily meetings sometimes frustrate team members

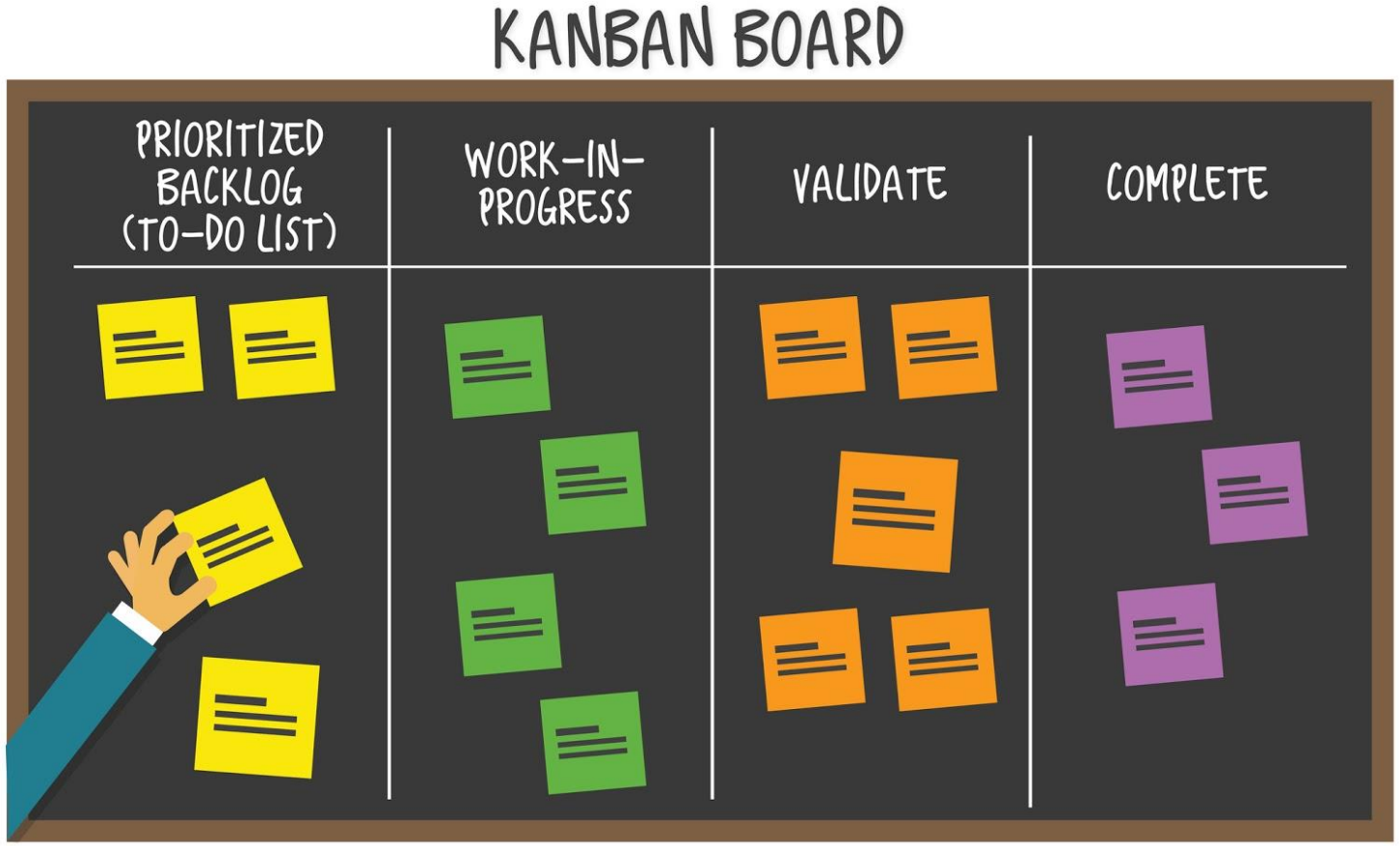
2- Kanban

- It is a visual framework used to implement Agile and shows what to produce, when to produce it, and how much to produce.
- It encourages small, incremental changes to your current system and does not require a certain set up or procedure.

2- Kanban

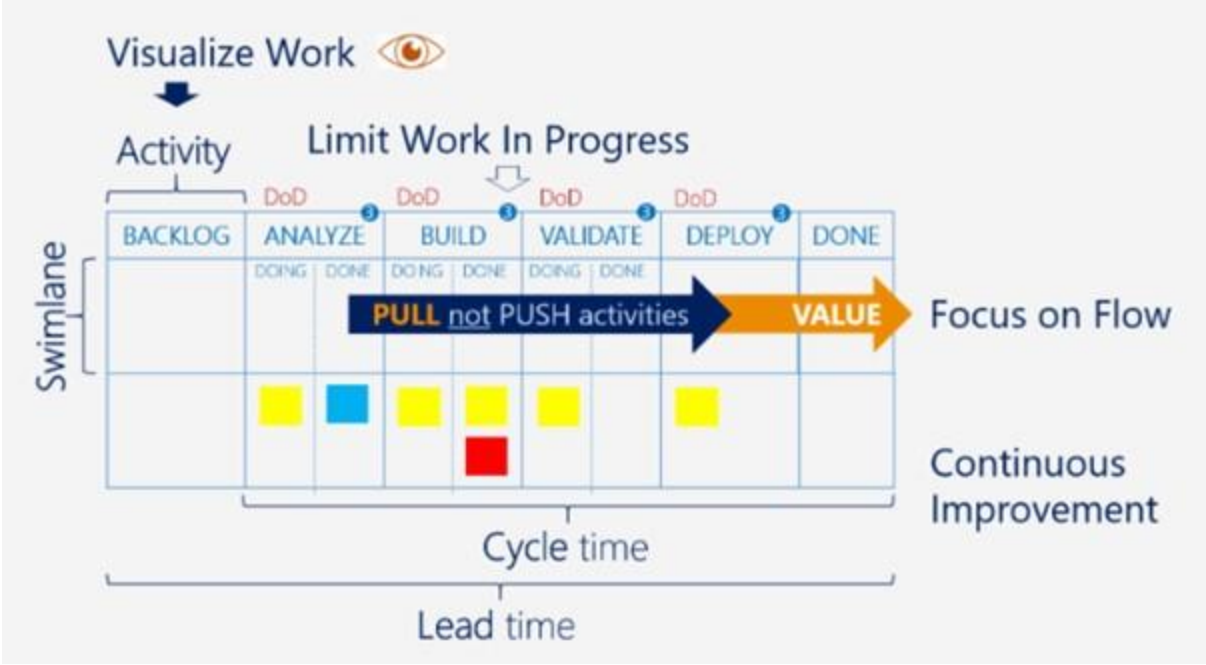
- The simplest visualization of workflow with 4 different states:
    - Prioritized backlog
    - In progress
    - Testing
    - Done

2- Kanban Tool



2- What Are the Kanban Practices?

1. Visualization of workflow
2. Limiting work in progress (WIP)
3. Managing flow
4. Making process policies explicit
5. Feedback loops
6. Improving collaboratively, evolving experimentally

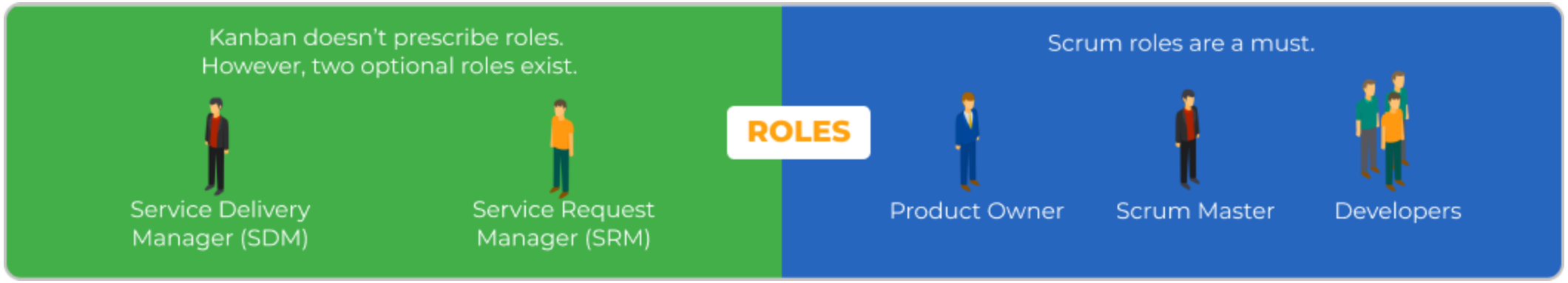


2-Advantages and Disadvantages of Kanban

Advantages	Disadvantages
Ease of use	Teams can overcomplicate the board
Reduces costs and wastage	Does not fit into a dynamic environment
Improved delivery speed	Lack of timing
Increased customer satisfaction	
Alignment between goals and execution	
Collaboration	

2- Kanban vs. Scrum

Kanban vs. Scrum: Roles and Accountabilities



2- Kanban vs. Scrum

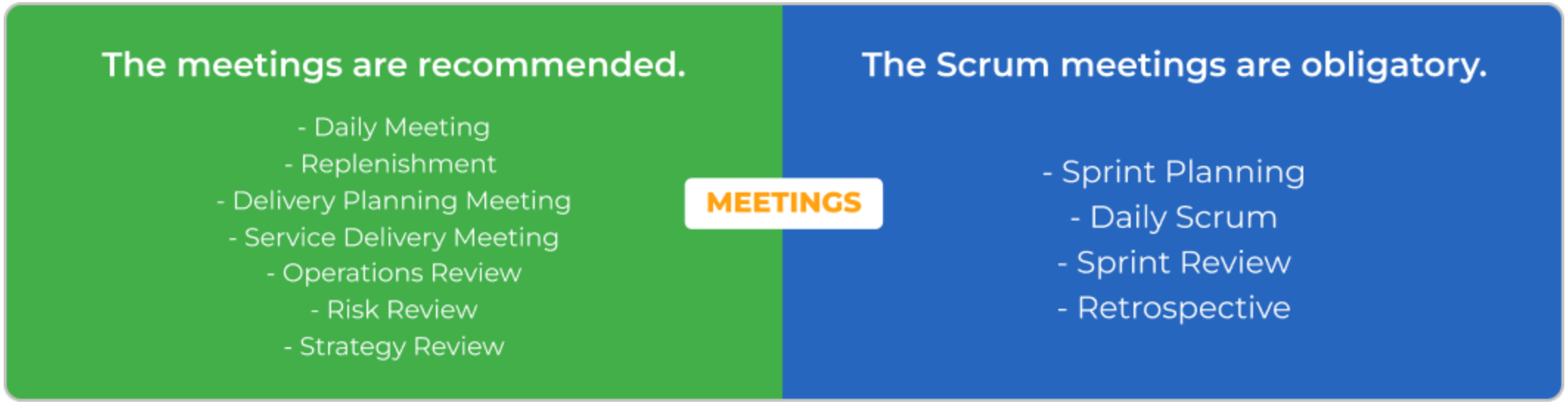
Kanban vs. Scrum: Planning

2- Kanban vs. Scrum

Kanban vs. Scrum: Work Commitment

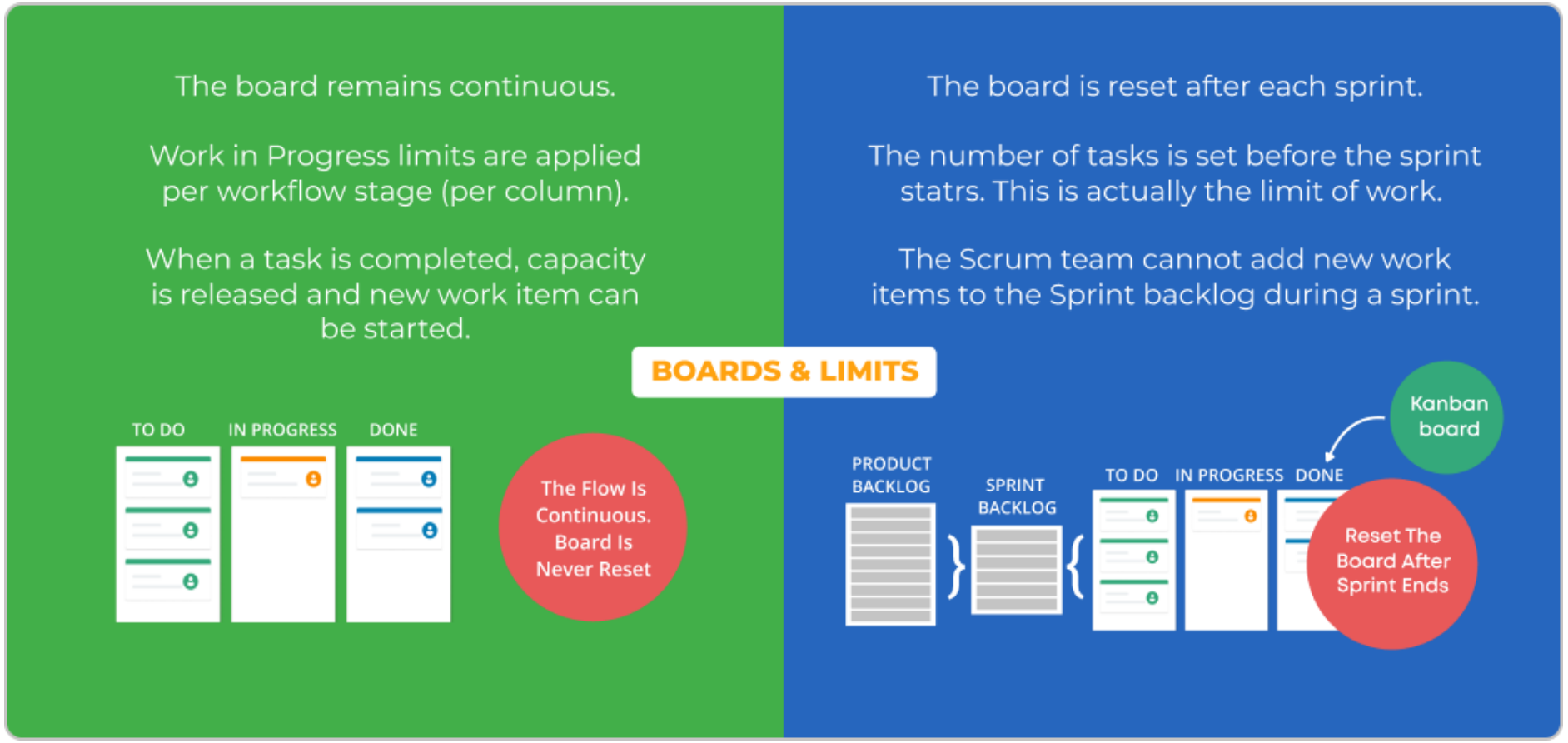
2- Kanban vs. Scrum

Kanban vs. Scrum: Meetings and Events



2- Kanban vs. Scrum

Kanban Board vs. Scrum Board



2- Kanban vs. Scrum

Kanban vs. Scrum: Core Key Performance Indicators (KPIs)

Cycle time begins when the new arrival enters the “in progress” stage, and somebody is actually working on it.
Lead time is the period between a new task’s appearance in your workflow and its final departure from the system.

2- Summery on Kanban vs. Scrum

	Kanban	Scrum
Nature	Kanban is an adaptive method	Scrum is a prescriptive framework
Principles	1. Start with what you do now 2. Agree to pursue evolutionary change 3. Encourage acts of leadership at all levels 4. Focus on customer's needs 5. Manage the work 6. Regularly review the network of services	1. Empiricism 2. Transparency 3. Inspection 4. Adaptation
Cadences	- Team-level cadences - Service-oriented cadences	- Sprint with a fixed length - Sprint planning - Daily Scrum - Sprint Review - Sprint Retrospective
Roles	- Service Delivery Manager* - Service Request Manager* (*no pre-defined roles are required)	- Product Owner - Scrum Master - Development Team
Metrics	- Cycle Time - Throughput - Work In Progress	- Velocity - Planned Capacity

3- Extreme Programming (XP)

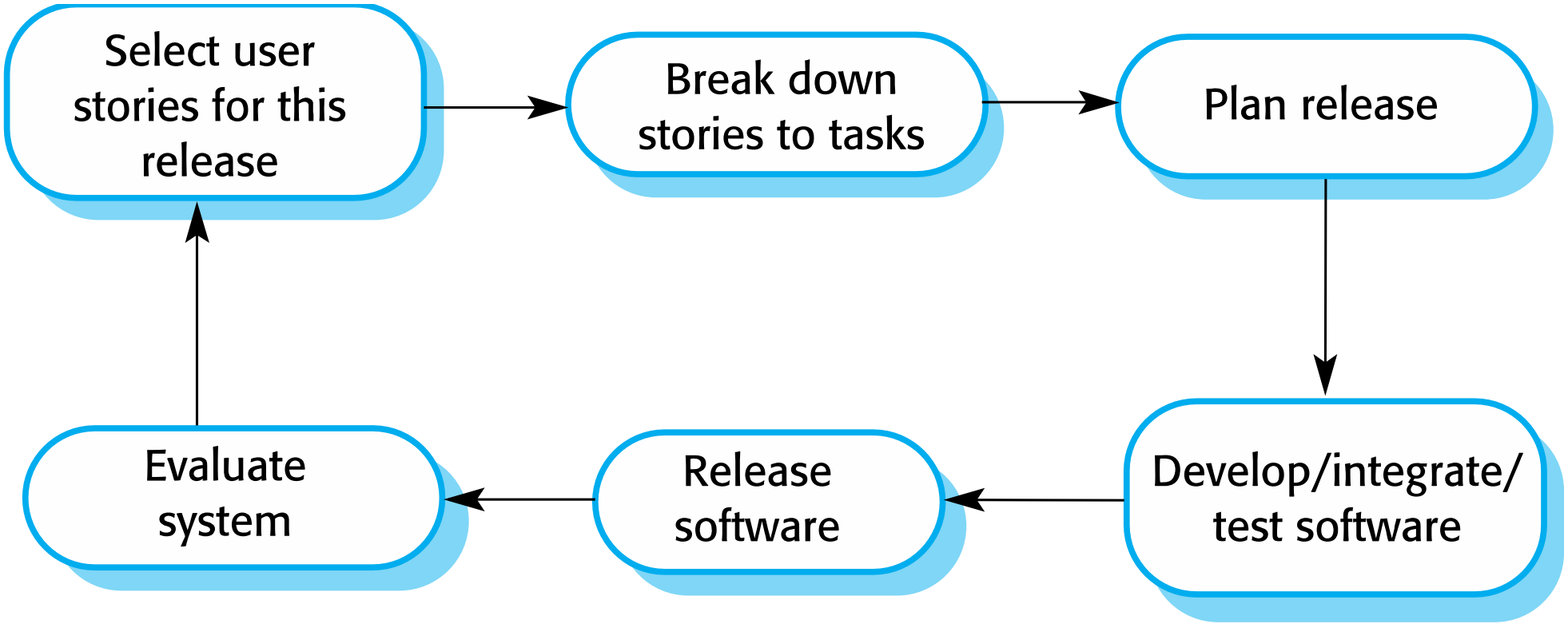
“ XP is a lightweight methodologies for small to medium sized teams developing software in the dace of vague or rapidly changing requirements”

- Kent Beck

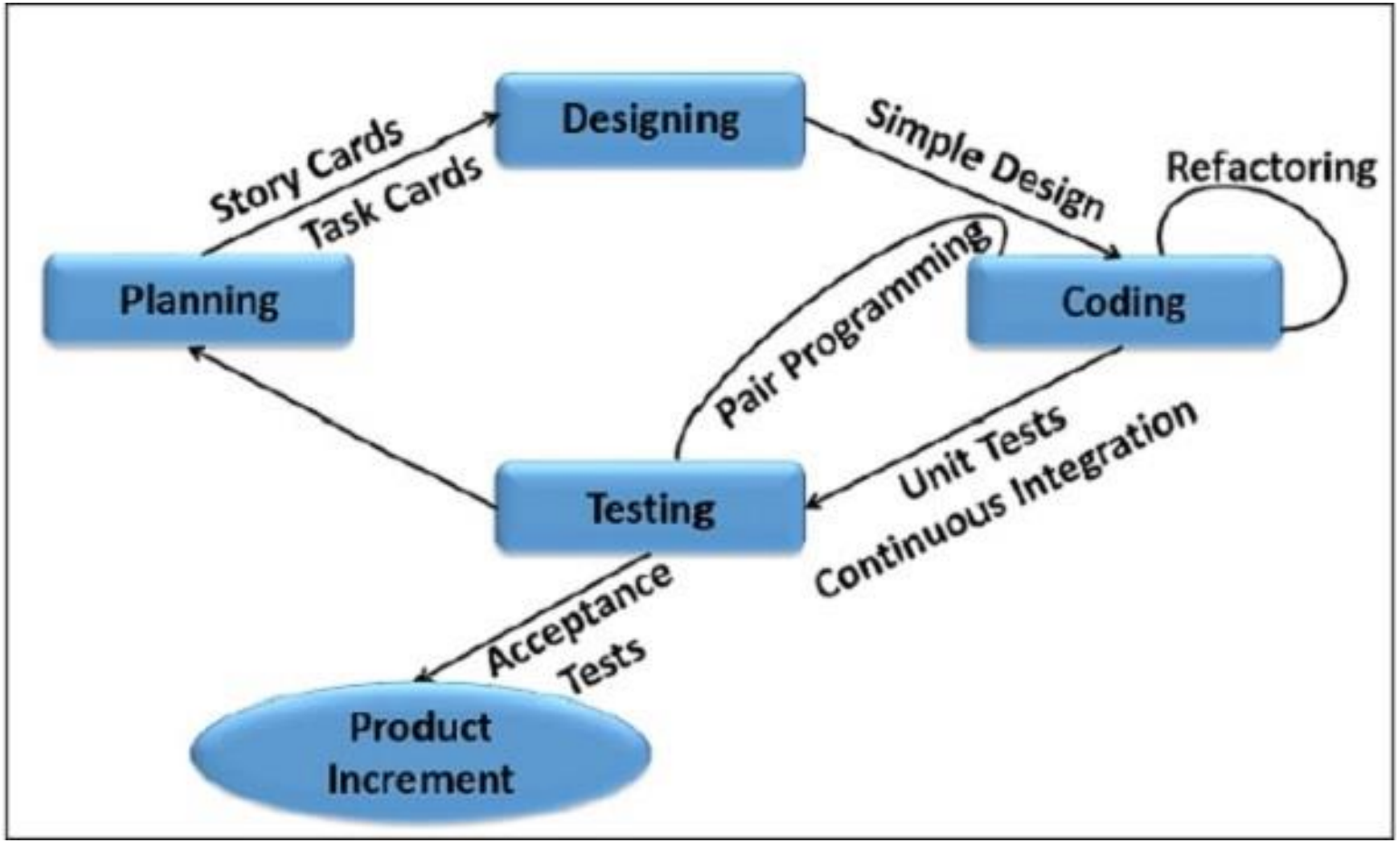
3- Extreme Programming (XP)

- Extreme Programming (XP) takes an ‘extreme’ approach to iterative development.
  - New versions may be built several times per day;
  - Increments are delivered to customers every 2 weeks;
  - All tests must be run for every build and the build is only accepted if tests run successfully.

3- Extreme Programming (XP)



3- Extreme Programming (XP)



3- Extreme Programming (XP)

- Key practices
  - User stories for specification
  - Refactoring
  - Test-first development
  - Pair programming

3- Extreme Programming (XP)

- Key practices
  - User stories for specification:
    - In XP, a customer or user is part of the XP team and is responsible for making decisions on requirements.
    - User requirements are expressed as scenarios or user stories.
    - These are written on cards and the development team break them down into implementation tasks. These tasks are the basis of schedule and cost estimates.
    - The customer chooses the stories for inclusion in the next release based on their priorities and the schedule estimates.
  - Refactoring
  - Test-first development
  - Pair programming

3- Extreme Programming (XP)

Prescribing medication

The record of the patient must be open for input. Click on the medication field and select either 'current medication', 'new medication' or 'formulary'.

If you select 'current medication', you will be asked to check the dose; If you wish to change the dose, enter the new dose then confirm the prescription.

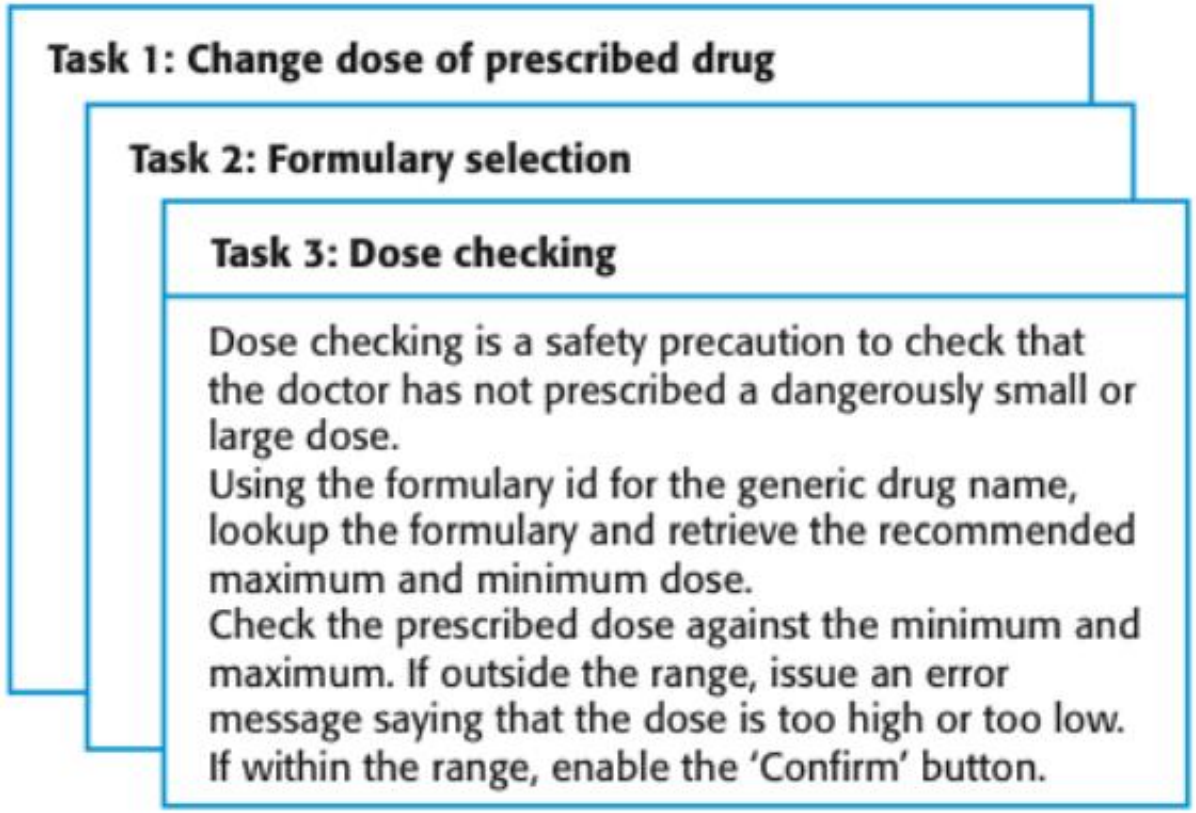
If you choose, 'new medication', the system assumes that you know which medication you wish to prescribe. Type the first few letters of the drug name. You will then see a list of possible drugs starting with these letters. Choose the required medication. You will then be asked to check that the medication you have selected is correct. Enter the dose then confirm the prescription.

If you choose 'formulary', you will be presented with a search box for the approved formulary. Search for the drug required then select it. You will then be asked to check that the medication you have selected is correct. Enter the dose then confirm the prescription.

In all cases, the system will check that the dose is within the approved range and will ask you to change it if it is outside the range of recommended doses.

After you have confirmed the prescription, it will be displayed for checking. Either click 'OK' or 'Change'. If you click 'OK', your prescription will be recorded on the audit database. If you click 'Change', you reenter the 'Prescribing medication' process.

3- Extreme Programming (XP)



3- Extreme Programming (XP)

- Key practices
  - User stories for specification
  - Refactoring:
    - Programming team look for possible software improvements and make these improvements even where there is no immediate need for them.
    - This improves the understandability of the software and so reduces the need for documentation.
    - Changes are easier to make because the code is well-structured and clear.
    - However, some changes requires architecture refactoring and this is much more expensive
  - Test-first development
  - Pair programming

3- Extreme Programming (XP)

- Re-organization of a class hierarchy to remove duplicate code.
- Tidying up and renaming attributes and methods to make them easier to understand.
- The replacement of inline code with calls to methods that have been included in a program library

3- Extreme Programming (XP)

- Key practices
  - User stories for specification
  - Refactoring
  - Test-first development(XP testing features):
    - Incremental test development from scenarios.
    - User involvement in test development and validation.
    - Automated test harnesses are used to run all component tests each time that a new release is built.
  - Pair programming

3- Extreme Programming (XP)

- Tests are written as programs rather than data so that they can be executed automatically. The test includes a check that it has executed correctly.
- All previous and new tests are run automatically when new functionality is added, thus checking that the new functionality has not introduced errors.

3- Extreme Programming (XP)

- Key practices
  - User stories for specification
  - Refactoring
  - Test-first development
  - **Pair programming:**
    - Pair programming involves programmers working in pairs, developing code together.
    - This helps develop common ownership of code and spreads knowledge across the team.
    - It serves as an informal review process as each line of code is looked at by more than 1 person.
    - It encourages refactoring as the whole team can benefit from improving the system code.

3- Advantages and Disadvantages of Programming (XP)

Advantages	Disadvantages
Close contact with the customer	Provides additional work
Changes can be made at short notice	Requires team members to be self-disciplined
Code is clear and comprehensive	No proper documentation is produced constantly with ever requirement amendment/addition/removal

Questions: