Agile Software development

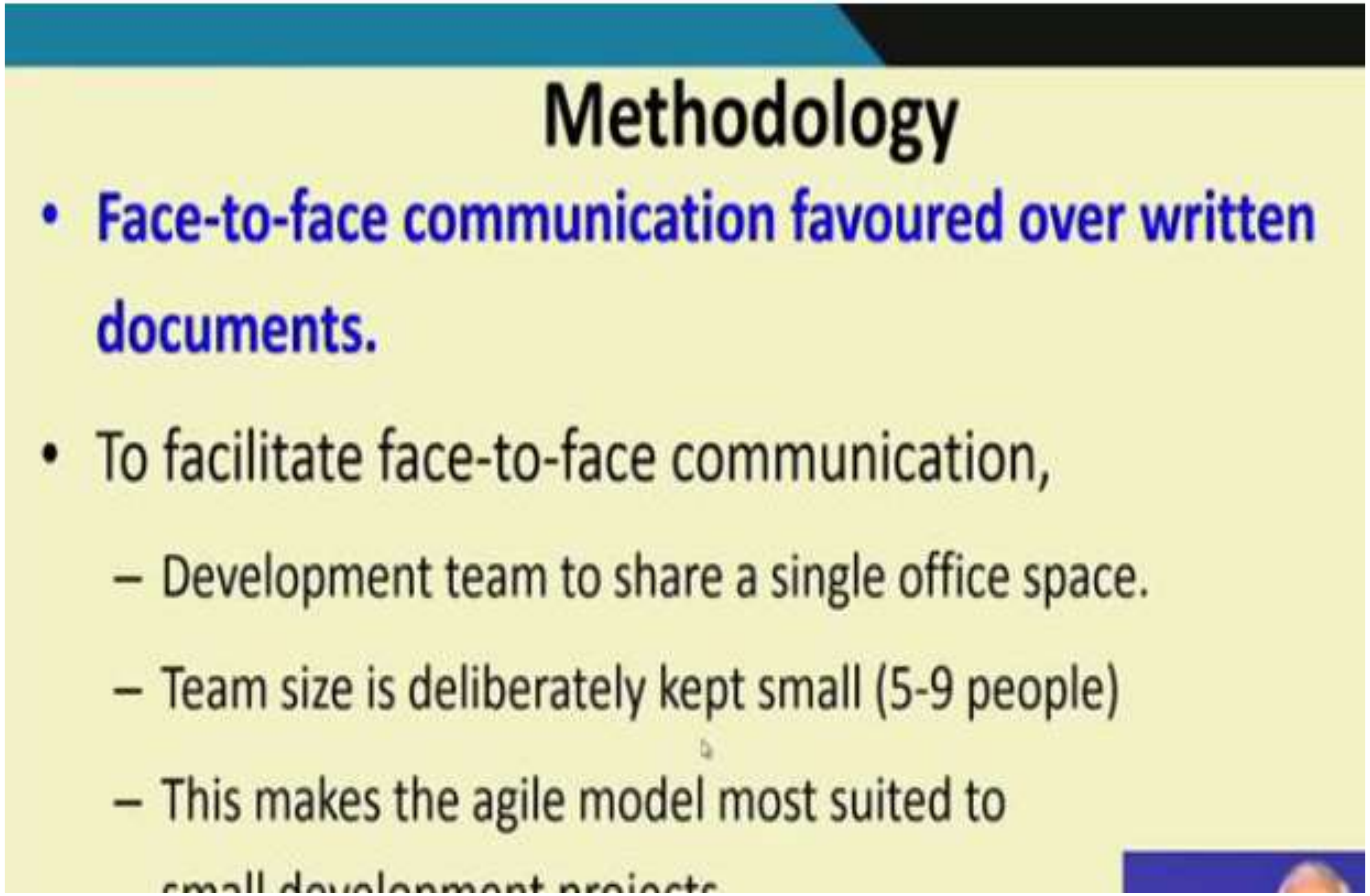
- The type of projects that are undertaken have undergone a drastic change over the years.
- Now the projects are of very short duration; 1 month, 2 month or 3 month projects are very common whereas,
- earlier there were multiyear projects; 3 year, 4 year, 5 year projects and now lot of reuse is being made and there is only customization work.

Main techniques of agile Model

1. Here no formal requirements are developed; it is based on user stories; as the name says that these are more informal than a requirements specification. These are like stories; these are simpler than the use cases.
2. To give a overall design perspective, agile model proposes metaphors where there is a common vision of what is required and based on that the development starts. Wherever required a spike is done; a spike is a simple program that is written to explore potential solution. We can see that it is similar to a prototype. Wherever there is uncertainty develop a spike, check out the alternative whether the spike performs well and so on and also refactor.

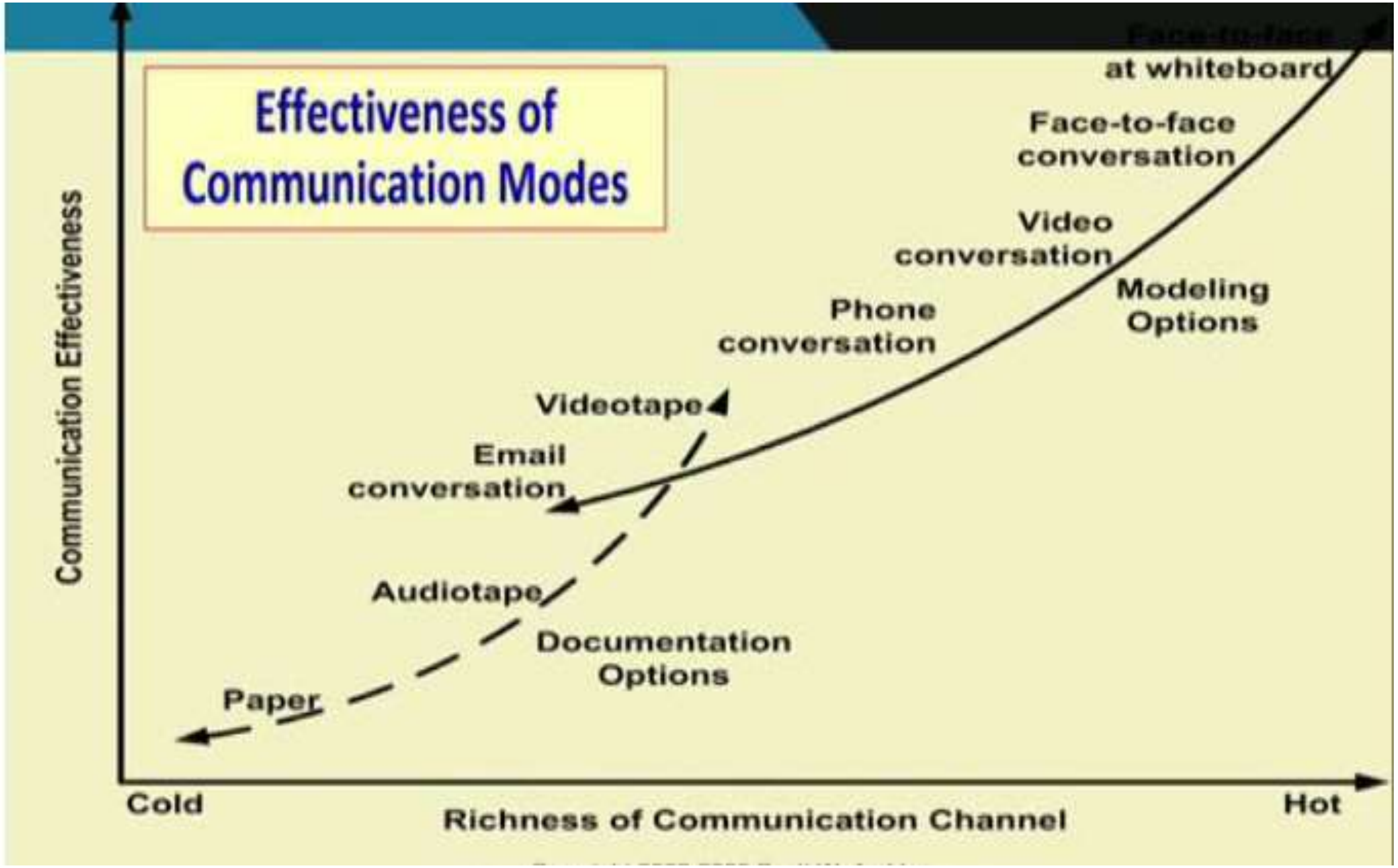
- Agile model is incremental, each time one increment is planned developed and deployed at the customer site, no long term plans are made. Iteration may not add significant functionality, it may just only enhance the existing functionality.

- But then at the end of each iteration invariably, code is deployed at the customer site and the length of the iteration is usually fixed something like 2 to 4 weeks. After each iteration, the customer puts the code into regular use. It is not that they just evaluate it, they just they start using it regularly

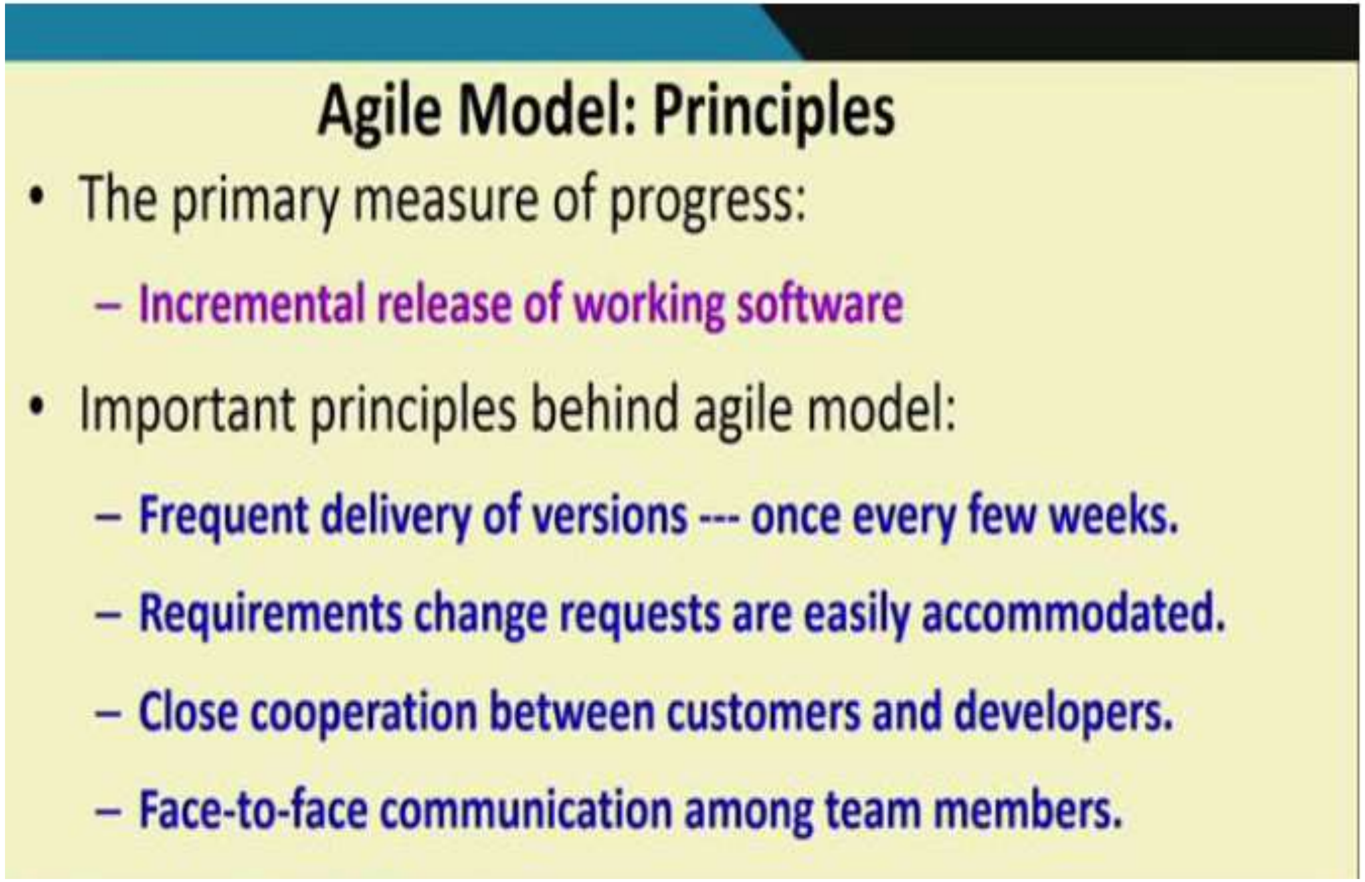


Methodology

- **Face-to-face communication favoured over written documents.**
- To facilitate face-to-face communication,
  - Development team to share a single office space.
  - Team size is deliberately kept small (5-9 people)
  - This makes the agile model most suited to small development projects

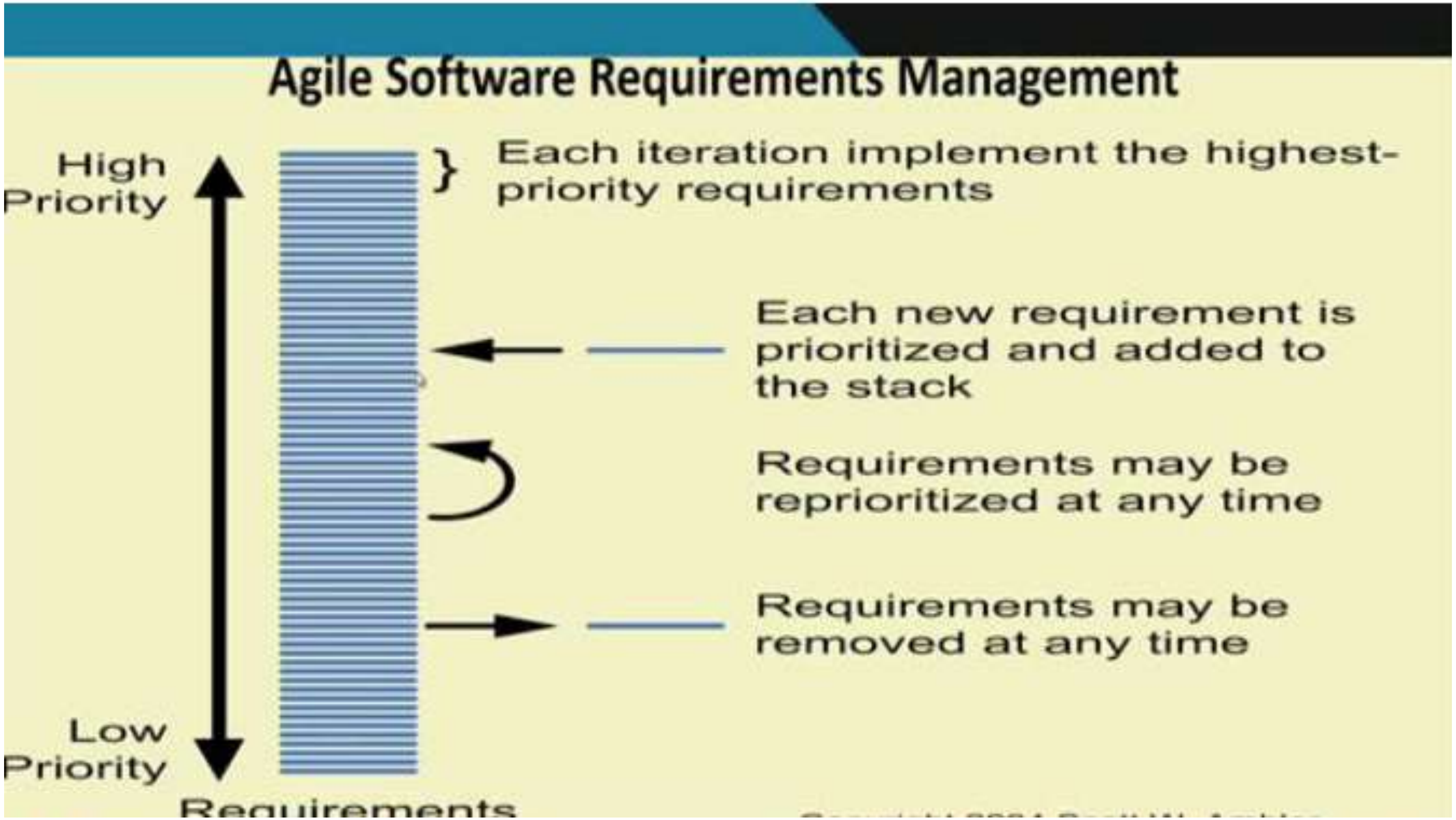


Effectiveness of Communication Modes



Agile Model: Principles

- The primary measure of progress:
  - Incremental release of working software
- Important principles behind agile model:
  - Frequent delivery of versions --- once every few weeks.
  - Requirements change requests are easily accommodated.
  - Close cooperation between customers and developers.
  - Face-to-face communication among team members.



Agile Software Requirements Management

Each iteration implement the highest-priority requirements

Each new requirement is prioritized and added to the stack

Requirements may be reprioritized at any time

Requirements may be removed at any time

Pitfalls of the agile model

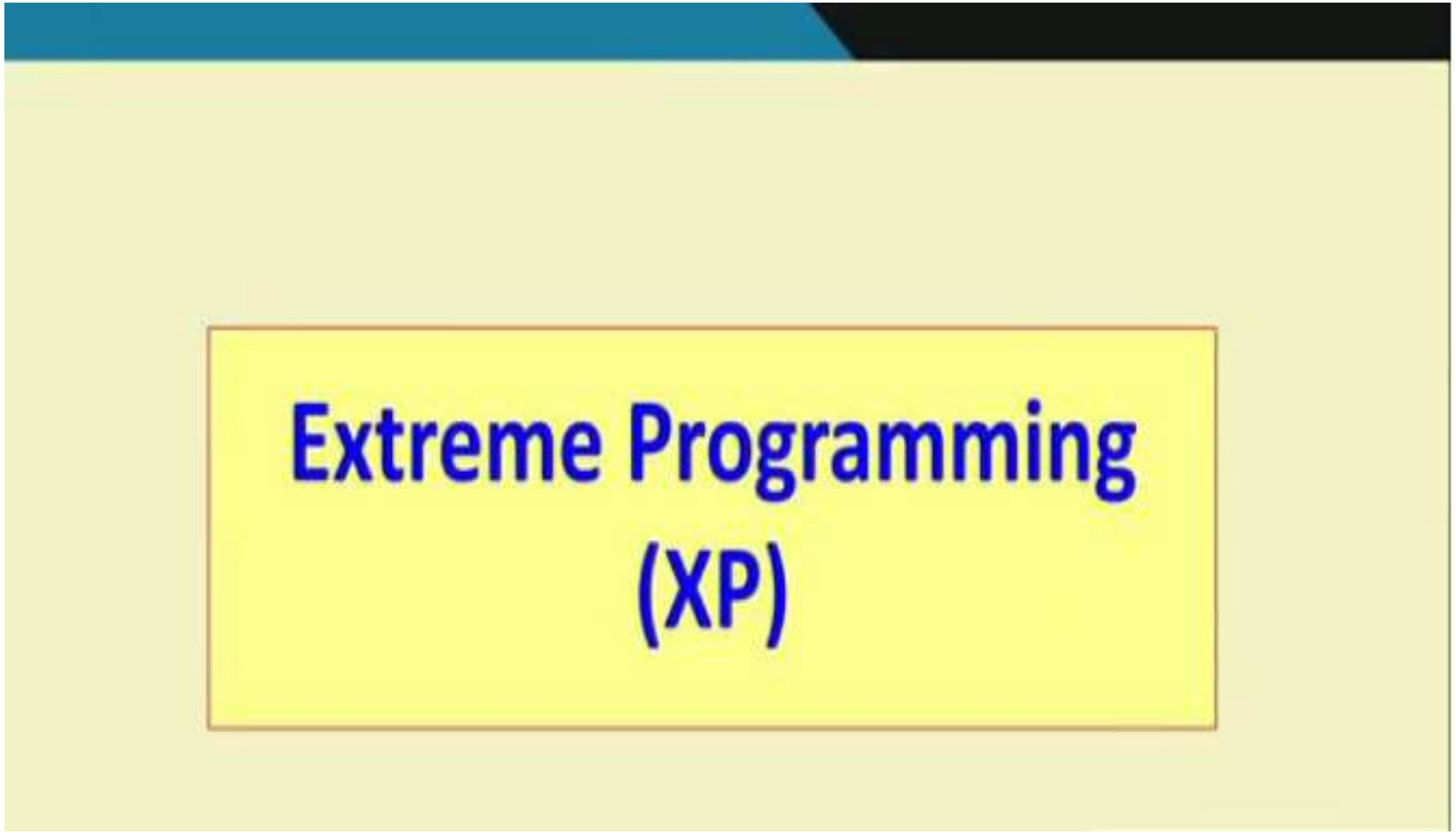
- First thing is that here documents are not there. It is based on explanation on a whiteboard and so on. And therefore, somebody can misunderstand.

- Doing away with documentation is a good thing as it saves time and effort of preparing documents but then lack of documentation can make maintenance difficult after the project completes and the team disperses

- Feature creep: The customers or the developers become more ambitious. They just keep on thinking new features which can be incorporated without thinking of whether how much the feature will be used; what will be the value to the customer, what will be the cost of developing it, feature after feature get added. So, there is a higher risk of feature creep in the agile model because the freedom is given to customer for feedback .

- And therefore, it becomes difficult for the project manager to manage the feature creep and also since the development is dynamic, the features change new features get added and so on.

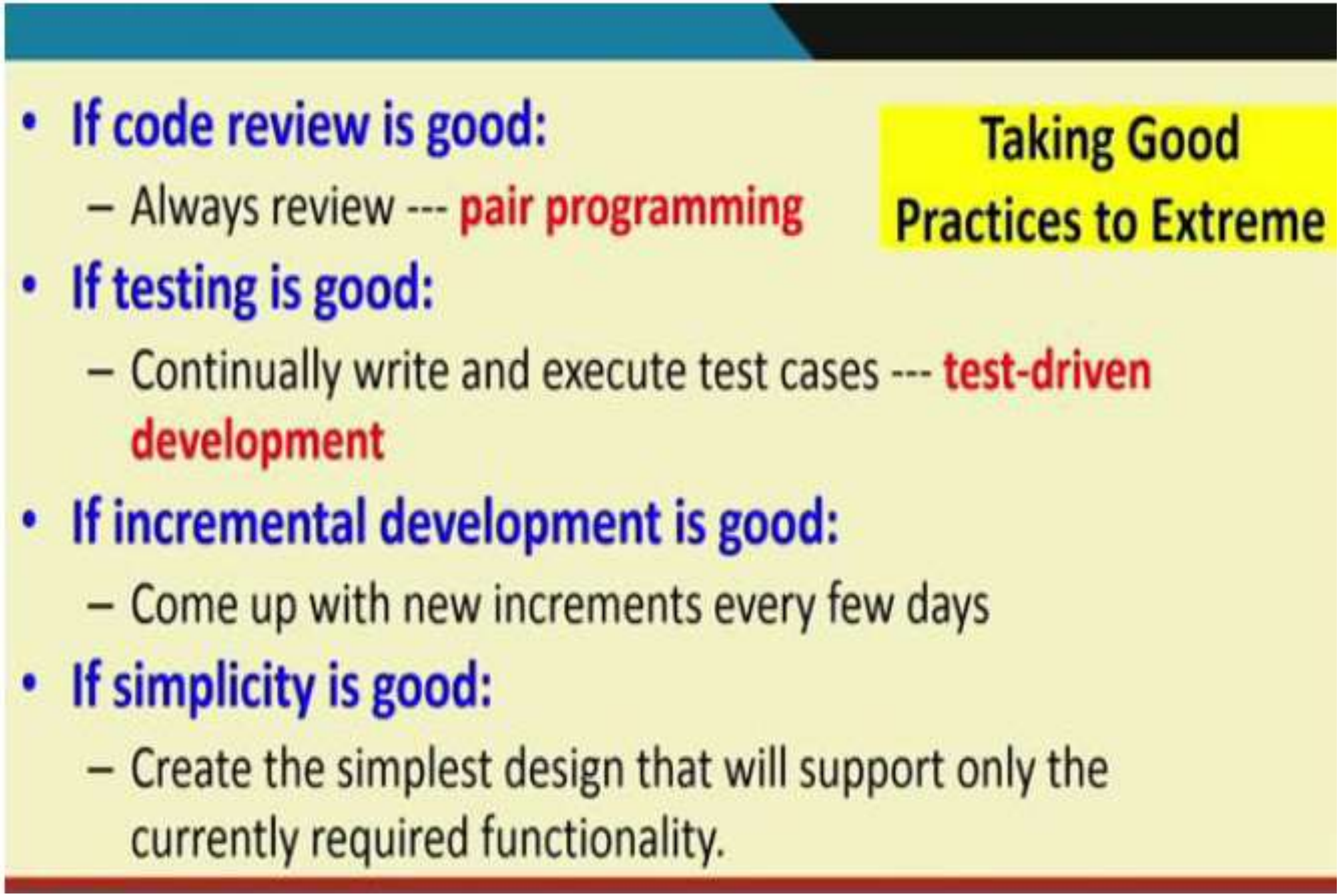
- To give an upfront cost to a development becomes very difficult; to give a upfront timeline by which development will be done is difficult and quality difficult to assure .



Extreme Programming (XP)

- With this overall understanding of the agile model, let us look at some specific agile processes. One of the agile process which is very popular is the Extreme Programming which is also known as the XP model.

- It was proposed by Kent Beck, 1999.



Taking Good Practices to Extreme

- **If code review is good:**
  - Always review --- **pair programming**
- **If testing is good:**
  - Continually write and execute test cases --- **test-driven development**
- **If incremental development is good:**
  - Come up with new increments every few days
- **If simplicity is good:**
  - Create the simplest design that will support only the currently required functionality.

Pair Programming

- The main items that are introduced here is pair programming. In a pair programming, each desktop is manned by two programmers. The main idea here is that reviewing is a good thing. When one programmer writes the program, the other programmers goes through the code and reviews it give suggestions how to make it better code, more efficient code, avoid mistakes and so on

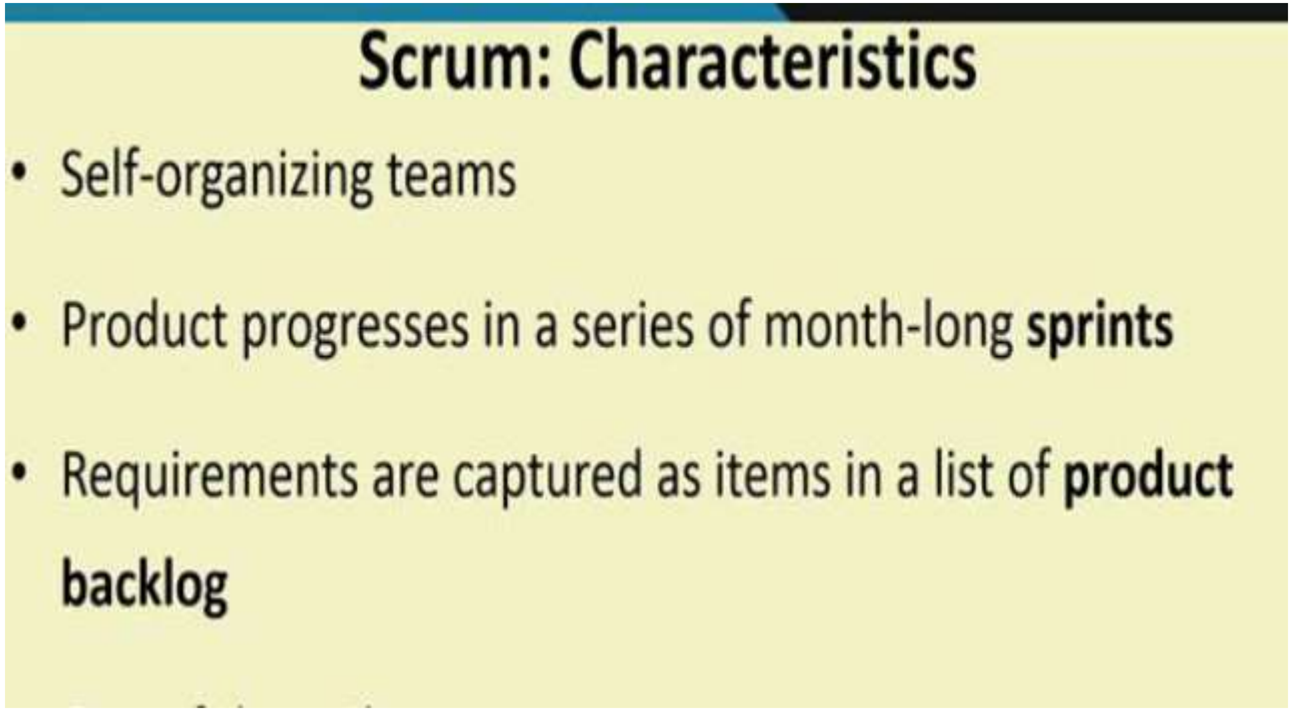
Test Driven Development (TDD)

- . Before a feature is implemented, the test cases are written for that feature and the feature is considered passed when it passes the test cases. This is called as test driven development.

- Here every day the testing takes places unlike the waterfall model where testing is at the end. Here the test cases are written continually and the test cases are executed before a feature is passed

- Now, let us look at another agile development process which is called a scrum

Scrum



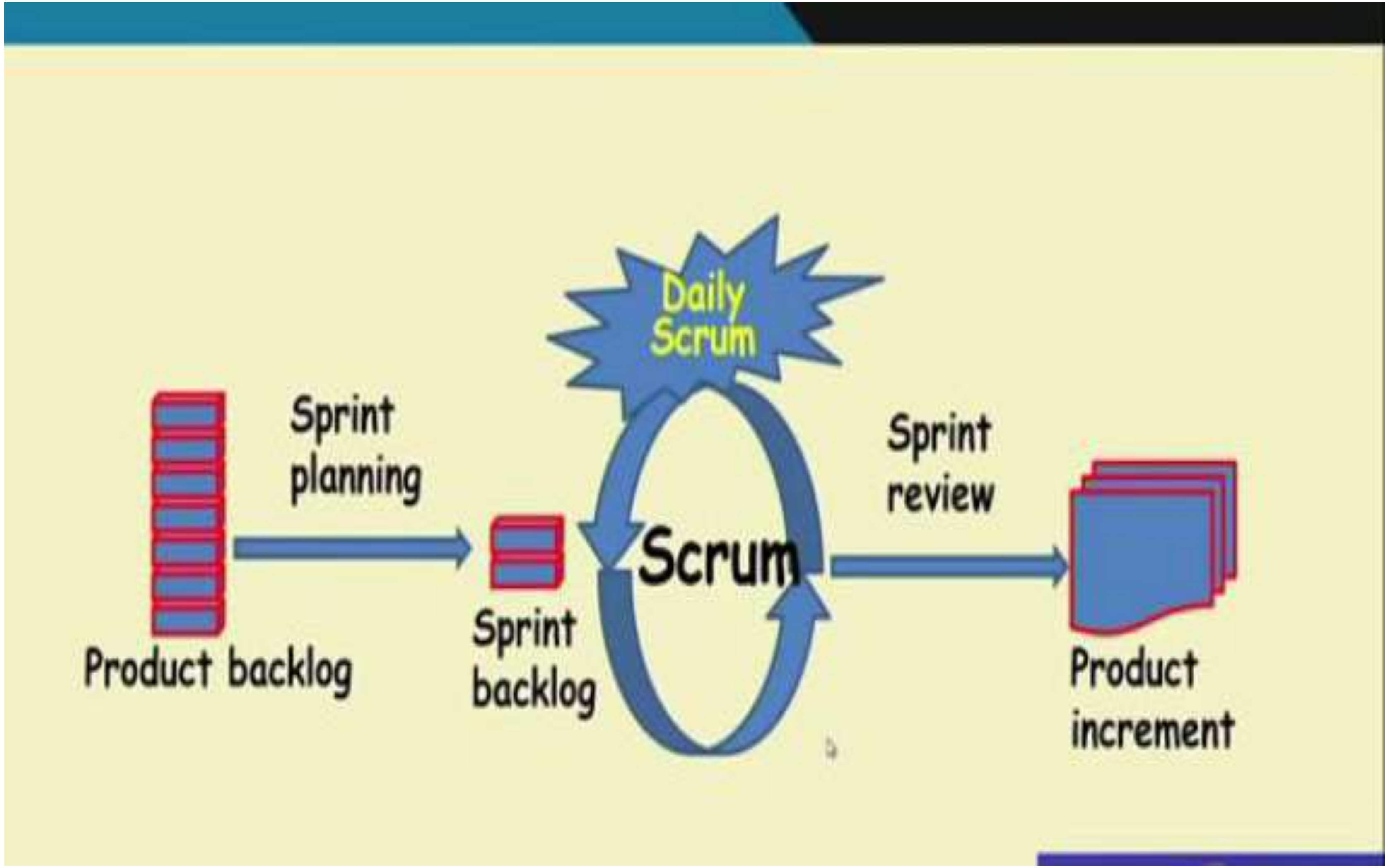
Scrum: Characteristics

- Self-organizing teams
- Product progresses in a series of month-long **sprints**
- Requirements are captured as items in a list of **product backlog**

Characteristics of scrum

1. The main characteristic of scrum is self organizing teams that is the team members decide among themselves who will do what, who will do testing, who will do which function development etcetera.

2. Here the product progresses in a series of month long sprints; the sprints are basically increments. So, the increments are called here as sprints typically one month. Here the requirements stack is called as a product backlog.



Daily Scrum

Sprint planning

Product backlog

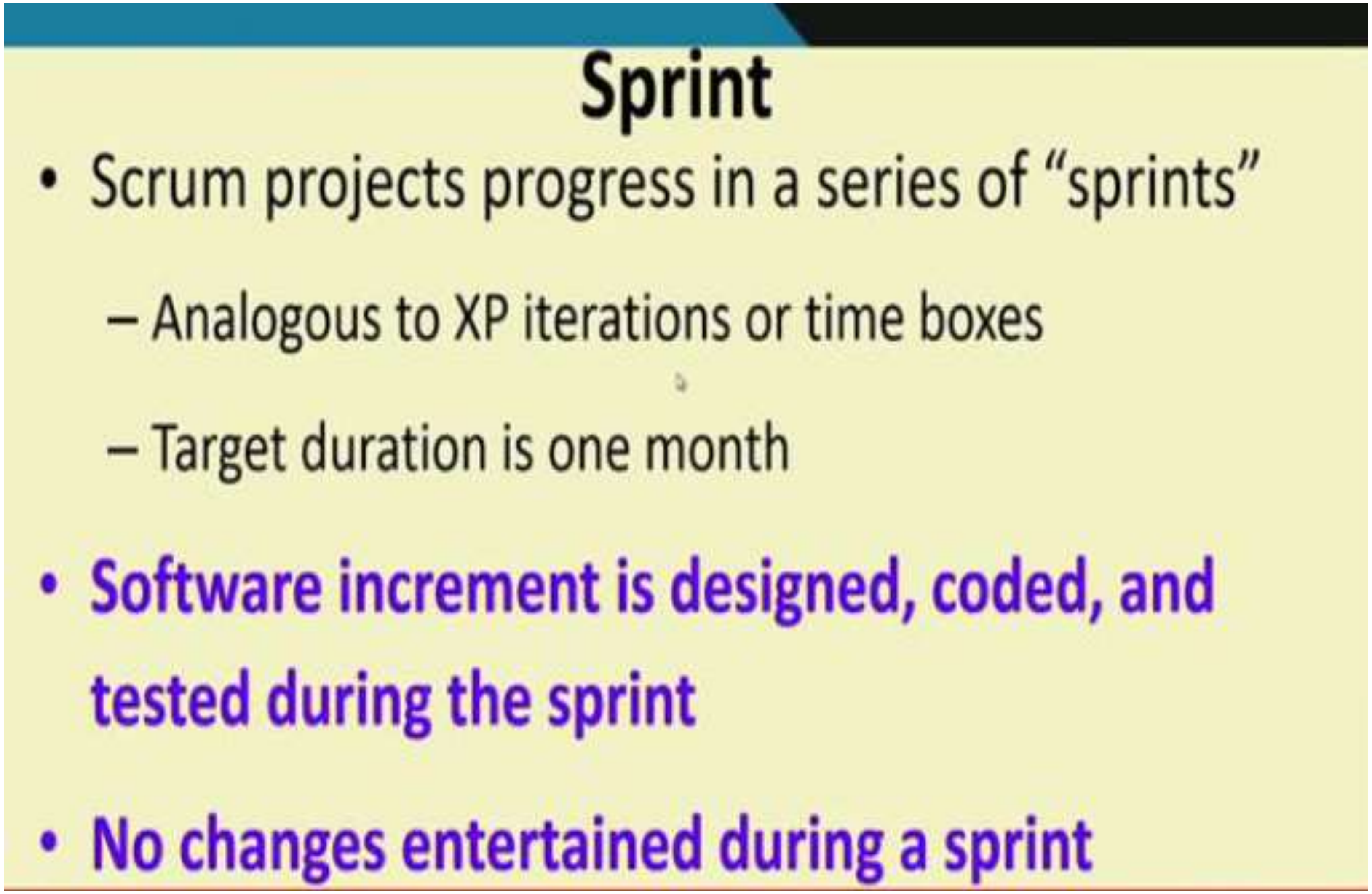
Sprint backlog

Scrum

Sprint review

Product increment

- Product Backlog is the requirements that have been gathered which arise as the development proceed some may get deleted, changed and so on
- . In each sprint, a sprint is a month long activity. One of the top priority feature requirement is taken out that forms the sprint backlog and once the sprint backlog is obtained, this is not changed anymore. It is immune to change and this is developed over a month long iteration here and every day the developers meet for a daily scrum meeting to review what has happened, what is the next thing to do and so on and they complete the sprint backlog.
- In the sprint review; the sprint is reviewed and the product increment is deployed at the customer site.



Sprint

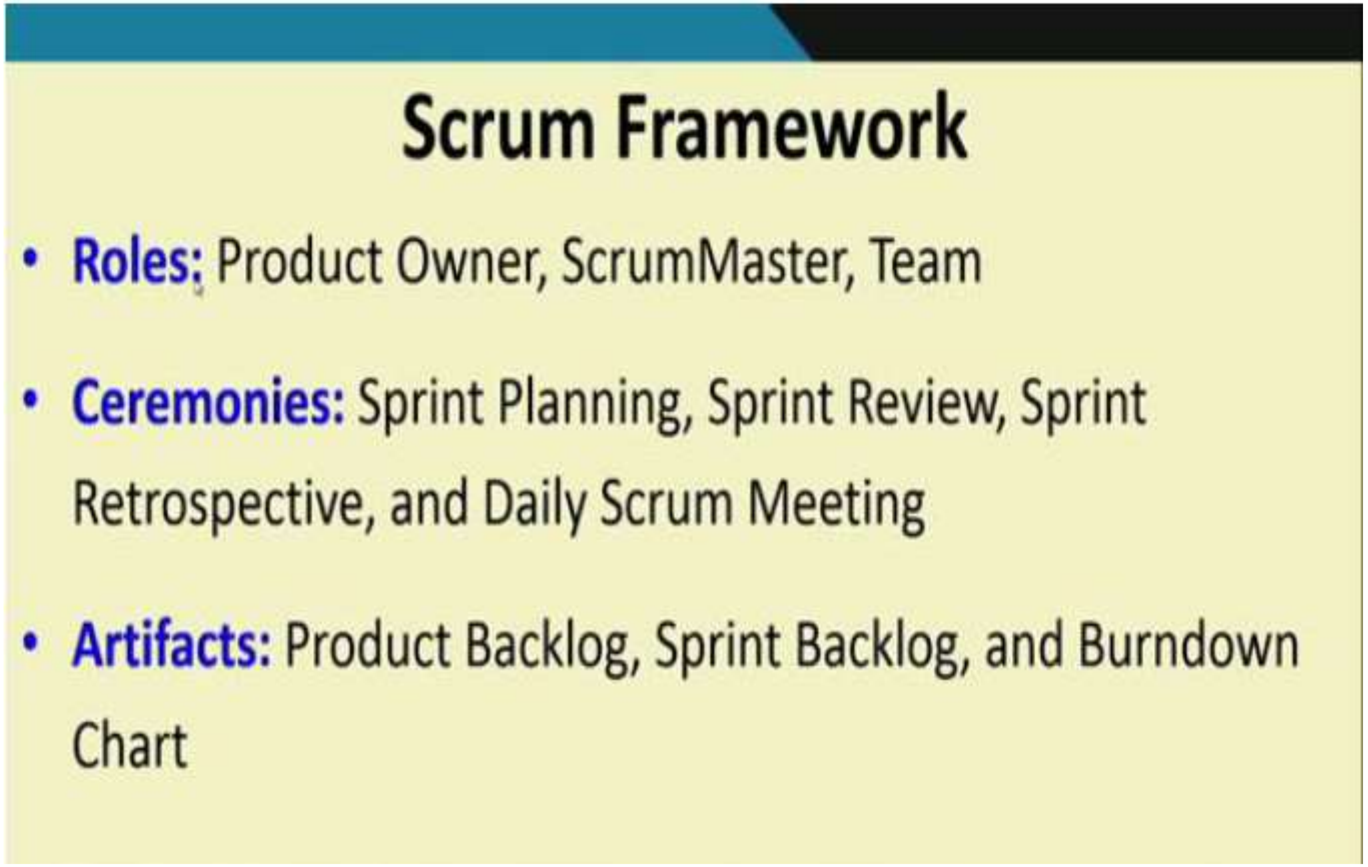
- Scrum projects progress in a series of “sprints”
  - Analogous to XP iterations or time boxes
  - Target duration is one month
- **Software increment is designed, coded, and tested during the sprint**
- **No changes entertained during a sprint**

- The progress is in the form of sprints, duration is typically one month.

- In one month, the some of the features from the product backlog are taken up; design, coded, tested.

- And once the sprint starts the requirements that have been taken up are not allowed to change otherwise the sprint will not converge.

- One of the principle here is that once the feature have been taken out from the product backlog, they are not allowed to change.



Scrum Framework

- **Roles:** Product Owner, ScrumMaster, Team
- **Ceremonies:** Sprint Planning, Sprint Review, Sprint Retrospective, and Daily Scrum Meeting
- **Artifacts:** Product Backlog, Sprint Backlog, and Burndown Chart

Scrum Terminologies

- **Role**s
- product owner. One of the team member acts as on behalf of the product owner; that is a customer. He has the customer perspective,
- the scrum master is like a project manager and then the team members.

- There are various **ceremonies** that get conducted during development; one is the sprint planning, the sprint review at the end of a sprint, sprint retrospective and daily scrum meeting.
- There are various **artifacts** produced one is the product backlog which keeps track of the requirements that have been identified so far, more requirements can be identified as the development proceeds and these are kept in a prioritized order.
- The sprint backlog this is the activities to be done during the sprint.
- The burn down charts here how much progress has been made are depicted in the form of burn down charts.