What is Agile?

- Agile is the ability to create and respond to change. It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment.
- The authors of the Agile Manifesto chose “Agile” as the label for this whole idea because that word represented the adaptiveness and response to change which was so important to their approach.
- It’s really about thinking through how you can understand what’s going on in the environment that you’re in today, identify what uncertainty you’re facing, and figure out how you can adapt to that as you go along.

Why Agile?

Agile methodology has become increasingly popular in recent years due to its ability to help teams become more flexible, collaborative, and adaptable to change. Some reasons why organizations adopt Agile include:

1. Faster time-to-market: Agile enables teams to produce working software or products in short iterations, allowing for quicker feedback and faster time-to-market.
2. Improved quality: Agile prioritizes continuous improvement, which helps teams to identify and address issues quickly, leading to improved quality.
3. Increased customer satisfaction: Agile places a strong emphasis on customer collaboration, ensuring that the team is building products that meet customer needs and expectations.
4. Greater flexibility: Agile allows teams to adapt to changing requirements and priorities, which is essential in today's fast-paced business environment.
5. Enhanced team collaboration: Agile promotes collaboration and teamwork, which helps to build a strong team culture and improve overall productivity.

6. Better project visibility: Agile encourages transparency and communication, which helps team members to stay informed about project progress, potential roadblocks, and other important details.

Overall, Agile methodology provides a framework for teams to work together more effectively and efficiently, leading to better results and increased customer satisfaction.

AGILE PHILOSOPHY

Agile philosophy is a set of principles and values that originated in software development but has been applied across a variety of industries. It emphasizes a collaborative and iterative approach to problem-solving and project management, with a focus on delivering working software or products in short iterations. The Agile philosophy prioritizes customer collaboration, flexibility, and responsiveness to change, and values people and communication over processes and tools. Agile teams are typically self-organizing and cross-functional, with a strong emphasis on continuous improvement and reflection. Agile methodologies like Scrum, Kanban, and Extreme Programming (XP) provide frameworks and guidelines for implementing Agile principles in practice. Overall, the Agile philosophy has proven to be an effective way to manage complex projects and adapt to rapidly changing business environments.

Agile Project Management

In the core of Agile project management lies the word "agility", which means "mobility, nimbleness", as well as from the Latin "agere": "to do, to act". This signifies the ability to move something forward in a quick way that allows easy changes of direction.

So, in terms of project management, **"agility" has five essential attributes that form the building blocks of the Agile process**:

- Transparency

- Customer focus
- Adaptability
- Sense of Ownership (Effective Leadership)
- Continuous Improvement

PRINCIPLE OF PROJECT MANAGEMENT SYSTEM

1. ***Satisfy Customers through Early & Continuous Delivery***

   By applying this concept, you will increase your process's agility and respond to changes in a timely fashion. On the other hand, your customers will be happier because they will get the value they are paying for more frequently. Also, they will be able to provide you with feedback early on, so you will be able to decrease the likelihood of making significant changes later in the process.

2. ***Welcome Changing Requirements Even Late in the Project***

   In traditional project management, any late-stage changes are taken with a grain of salt as this usually means scope creep and thus higher costs. However, Agile teams aim to embrace uncertainty and acknowledge that even a late change can still bear a lot of value to the end customer. Due to the nature of Agile's iterative process, teams shouldn't have a problem responding to those changes in a timely fashion

3. ***Deliver Value Frequently***

   Its prime goal is to reduce the batch sizes that you use to process work. This principle became necessary due to the extensive amounts of documentation that were part of the planning process in software development at the end of the 20th century. Logically, by taking it to heart, you will reduce the time frame for which you are planning

and spend more time working on your projects. In other words, your team will be able to plan in a more agile way

4. ***Break the Silos of Your Project***

   The goal is to create a synchronization between the people who create value and those who plan or sell it. This way, you can make internal collaboration seamless and improve your process performance.

5. ***Build Projects Around Motivated Individuals***

   The logic behind the fifth of the Agile principles is that by reducing micromanagement and empowering motivated team members, projects will be completed faster and with better quality.

6. ***The Most Effective Way of Communication is Face-to-face***

   With the development of technology, you can interpret this Agile principle from face-to-face to "synchronous" or otherwise direct communication. So as long as you have a way to quickly reach your team and discuss work matters without bouncing back and forward emails for days, you are good to go.

7. ***Working Software is the Primary Measure of Progress***

   The 7th of the Agile core principles is pretty straight forward. It doesn't matter how many working hours you've invested in your project, how many bugs you managed to fix, or how many lines of code your team has written. If the result of your work is not the way your customer expects it to be, you are in trouble

8. ***Maintain a Sustainable Working Pace***

   when putting Agile to practice, your goal is to avoid overburden and optimize the way you work so you can frequently deliver to the market and respond to change without requiring personal heroics from your team.

9. ***Continuous Excellence Enhances Agility***

In a development context, this principle allows teams to create not just working software but also a stable product of high quality. As a result, changes to the code will be less likely to impact bugs and malfunctions negatively. Still, the 9th of the Agile management principles is applicable in every industry. When you maintain operational excellence, you will have less trouble reacting to changes and maintaining agility.

10. ***Simplicity is Essential***

Your customers are not paying for the amount of effort you invest. They are buying a solution to a specific problem that they have. Keep that in mind, when implementing Agile and avoid doing something just for the sake of doing it.

11. ***Self-organizing Teams Generate Most Value***

12. ***Regularly Reflect and Adjust Your Way of Work to Boost Effectiveness***

Finally, we've come to the last of the Agile management principles. It is related to evaluating your performance and identifying room for improvement. The long version of the principle states:

The Agile Manifesto

We are uncovering better ways of developing software by doing it and helping others do it? Through this work we have come to value:

- ✓ Individuals and interactions over processes and tools.
- ✓ Working software over comprehensive documentation.
- ✓ Customer collaboration over contract negotiation.
- ✓ Responding to change over following a plan.

That is, while there is value in the items on the right, we value the items on the left more.

The following 12 Principles are based on the Agile Manifesto.

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer’s competitive advantage.

3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
9. Continuous attention to technical excellence and good design enhances agility.
10. Continuous attention to technical excellence and good design enhances agility.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Agile model - SCRUM

The **scrum** is the framework for implementation of agile software product development/project management. It can also be applied to software maintenance projects. The idea behind scrum is to divide the project into as many milestones as possible so that deliverables in incremental phases can be demonstrated to the customer much early and hence feedback/problems/ issues can be known at the early stage of the project life cycle. This small period of incremental phases in the agile process is called **sprint**. Sprint period is typically about 1 to 4 weeks. Usually it is taken as 1 month i.e. 4 weeks. But if the project is very small it can be taken as 2 weeks to have more milestones in entire duration as well as to have first milestone occur little earlier.

Scrum life cycle phases

Let us understand scrum framework for agile software development. The figure depicts scrum life cycle phases. Following steps or phases are followed in scrum based agile software development.

1.First the high level requirements are finalized based on end user perspective/opinions; hence it is called user stories. These user stories are broken up into tasks till we will not be able to break them down further. This is created in a document called as **product backlog** by **Product Owner** in association with customer. All the stories/ tasks are prioritized.

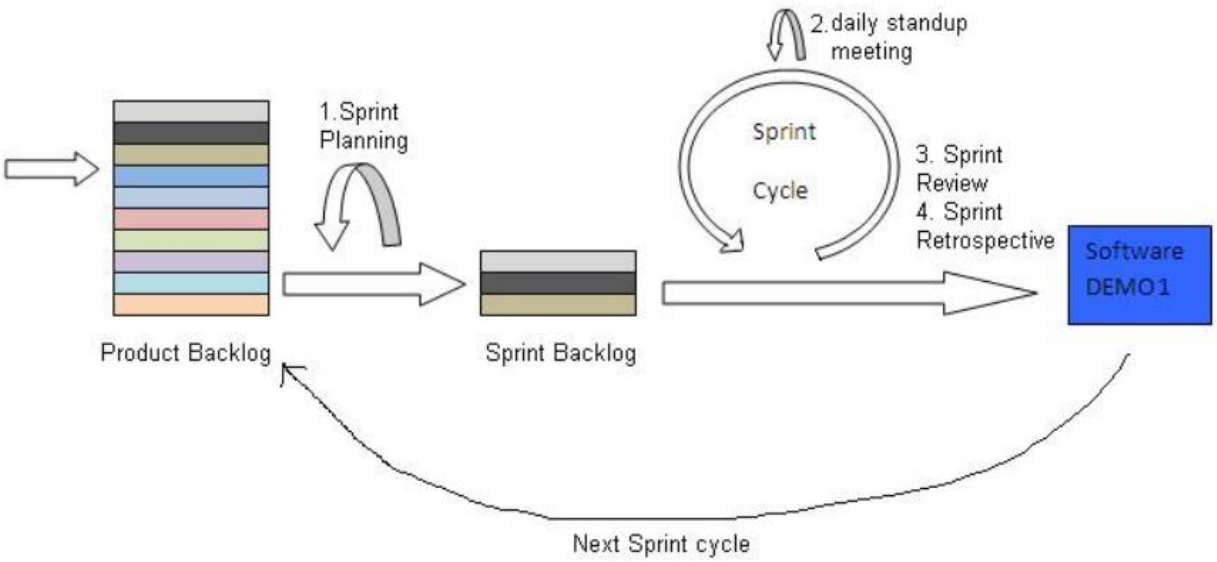
2.These tasks after prioritization are divided into sprints. Each sprint will contain set of tasks. This is done in sprint planning. The document created is called **sprint backlog**. The entire responsibility of sprint backlog lies with **Scrum team**.

3.Now the Sprint process will come in execution phase where in the scrum team will do mainly development/testing activities. During sprint execution phase scrum team daily meets to discuss the progress. These meetings are facilitated by **scrum master**. Hence scrum master plays the role of facilitator or servant leader. These meetings are called **daily standup meetings** or daily scrum meetings. These meetings are each of 15 minutes duration. In these meetings each team members will talk on three questions i.e. what they have achieved till yesterday, what they are going to do and are there any impediments on the way of smooth work progress.

4.After the end of sprint scrum team will do sprint review and also sprint retrospectives. This helps in better planning for next sprints.

5.After the sprint retrospectives the phase 1 or iteration 1 of the development carried out. It is evaluated with **definition of DONE** which is also called as acceptance criteria. Now the increment 1 of the development carried out will be demonstrated to the customer and feedback will be obtained from the customer. The same feedback will be utilized for next sprints.

6.The steps 1 to 5 mentioned above are revisited again in the consecutive sprints. Hence the **scrum life cycle** is a continuous process.



Scrum Life Cycle

Agile and Scrum Roles and Responsibilities

Entry-Level

Roles

Scrum Master: The Scrum Master is a highly experienced role responsible for ensuring that the Scrum process is followed. They are the facilitator of the Scrum team, and the Scrum master duties involve ensuring that the team is focused on delivering a product or service.

They are also responsible for removing obstacles that might affect the team's progress. Scrum masters need to have a deep understanding of the Scrum framework and must be adept at using it to ensure that the team is productive and efficient.

Responsibilities

- Understand the basics of Agile and Scrum, including sprints and user stories.
- Help to create sprint goals and objectives.
- Assist in the development of user stories and tasks.
- Help to track the progress of sprints and tasks.
- Work with stakeholders to ensure the team is meeting their goals.

Product Owner: The Product Owner oversees the entire product development process. They are the voice of the customer, and their job is to ensure that the product meets the user's needs.

The Product Owner requires an in-depth understanding of the customer's needs and the ability to translate those needs into a product that will satisfy those needs. They need a sound knowledge of the Scrum framework and its use to develop a product.

Responsibilities

- Act as a facilitator in sprint planning and review sessions.
- Help to define sprint goals and objectives.
- Lead the development of user stories and tasks.
- Manage sprints and tasks to ensure on-time delivery.
- Assume the role of a liaison between the team and stakeholders.
- Collaborate with product owners to ensure user stories and tasks are accurate and complete.
- Provide feedback and suggestions on ways to improve processes.

Senior-Level

Roles

Development Team Member: The Development Team Member is essential in the Scrum process. They are responsible for the implementation of the product and need to have a good understanding of the Scrum framework.

Team Members need to be able to collaborate with the other members of the team to ensure that the product is being developed most efficiently and effectively. They must also communicate effectively with the Product Owner and the Scrum Master to ensure that the product meets the customer's needs.

Stakeholder: The Stakeholder is a role responsible for representing the interests of the people affected by the product or service being developed. They must have a clear understanding of the goals and objectives of the development and an understanding of the Scrum process.

Responsibilities

- Take the lead in sprint planning and review sessions.
- Help to define sprint goals and objectives.
- Lead the development of user stories and tasks.
- Manage sprints and tasks to ensure on-time delivery.
- Assume the role of a liaison between the team and stakeholders.
- Guide product owners to ensure user stories and tasks are accurate and complete.
- Provide feedback and suggestions on ways to improve processes.
- Train and mentor team members on Agile and Scrum processes.

Scrum-Based Agile Ceremonies

The Scrum ceremonies are mostly applied at the team level. However, you can elevate them beyond that with the help of scaling frameworks such as LeSS (Large-Scale Scrum) or Scrum@Scale.

1. ***Daily Stand-Up***

✓ The daily stand-up meeting is perhaps the most famous and widely-applied Agile practice. It represents a 15-minute meeting

that is held standing up in front of an “information radiator” such as a Kanban board.

- ✓ In practice, it’s almost the same as the Daily Kanban meeting. The only difference is that the agenda of the meeting is stricter in Scrum as every team member should answer the following questions:

“What did I do yesterday?”
“What am I going to do today?”
“Is there something blocking my progress?”

- ✓ The people who attend the meeting include the development team and the Agile coach or Scrum master. The Product Owner can also attend but it’s not a requirement.

2. Sprint Planning

- ✓ The sprint planning ceremony in Agile is a 1-2 hours meeting that occurs at the start of every Scrum iteration/sprint. Its main intent is to populate a Sprint backlog and prepare a Sprint goal by committing to a batch of user stories from the Product backlog.
- ✓ During the meeting, teams choose the next highest priority items to execute in the sprint. This happens by analyzing their Agile velocity (how many story points they can execute in a single iteration).
- ✓ As part of the ceremony, the development team, the Product Owner and Scrum Master also write acceptance criteria for the user stories and prepare technical tasks for their execution.

3. *Sprint Review*

- ✓ Each iteration in Scrum ends with the Sprint Review. During the ceremony, Scrum teams present their finished increment (functionality) to the customer or the Product Owner. The intent of the ceremony is to get external feedback for their work/user stories, add new requirements to the Product backlog or reprioritize it, if necessary.
- ✓ Each Sprint Review involves the development team, the Scrum Master (Agile coach), and the Product Owner. It's good practice for the customer to attend the meeting too.

4. *Sprint Retrospective*

- ✓ To reflect on what has happened during the sprint, Scrum teams engage in Sprint Retrospectives right after the Sprint Review. The retrospective is a widely-applied Agile ceremony that Kanban teams often use too in an ad-hoc manner.
- ✓ During the event, teams analyze all problems or impediments that happened during the sprint and look for ways to prevent them from reoccurring. The goal of the retrospective is to continuously engage in "lessons learned" sessions over the course of the project.

5. *Backlog Refinement (Unofficial)*

- ✓ While being an unofficial Scrum-based Agile ceremony, the backlog refinement cadence is worth briefly mentioning. This meeting can happen towards the end of a sprint.
- ✓ Its main goal is to estimate, add more details, or break down the next high-priority user stories, so they're prepared for commitment during the Sprint Planning ceremony.

ENVISIONING PRODUCTS IN SCRUM

Envisioning Products in Scrum: Timing

Envisioning products in Scrum is an ongoing activity, not a one-time event. New or revised ideas about in-process products are also vetted through the envisioning process. As such, after the initial envisioning phase, subsequent envisioning sessions will likely be held to determine whether to proceed with the original vision, pivot to a new vision, or end the project (either by releasing the product as is or terminating the effort).

Envisioning Products in Scrum: Participants

During initial envisioning for a product in Scrum, the only required participant is the product owner. Typically, however, the product owner collaborates with one or more internal stakeholders, specialists in market research, business case development, user-experience design, and systems architecture. The process is shown in the image below. Optional participants and artifacts are indicated by a dashed outline

Notice that the Scrum team members are optional participants to envisioning. This reflects the reality that, during initial envisioning, a Scrum team might not yet be funded. Once product development is underway, or anytime a full Scrum team is available to lend insight and technical expertise, the Scrum team should be included.

Envisioning Products in Scrum: Process

For a new product, the main input to envisioning products in Scrum would be an idea that has cleared the organization's strategic filter. For an in-process product, the main input would be a pivoted idea.
A **pivoted idea in Scrum** is an idea that has been updated or revised based on user or customer feedback, funding changes, unpredictable moves by competitors, or other important changes that occur within the complex environment in which ideas must exist.

But these aren't the only inputs to envisioning products in Scrum. Organizations must also understand the following:

- **Planning horizon**: How far into the future to consider.
- Completion date for envisioning.
- Quantity and type of resources available to conduct envisioning.
- **Confidence threshold**: Set of information that the decision-makers need in order to make a go/no-go decision.

Envisioning products in Scrum requires several different activities, each generating an important output, such as the product vision, the initial product backlog, or (optionally) the product roadmap. Organizations should also perform any activities necessary to achieve the targeted confidence threshold in an economically sensible way.

THE PRODUCT VISION IN SCRUM

Organizations should create a compelling vision that represents the new product idea. This is not an elaborate, several-hundred page document. The **product vision** in Scrum is a brief statement of the desired future state that would be achieved by developing and deploying a product. A good vision should be simple to state and provide a coherent direction to the people who are asked to realize it.

An example of a good product vision is Kennedy's vision to go the the moon: "I believe that this nation should commit itself to achieving the goal before this decade is out of landing a man on the moon and returning him safely to Earth." In 31 words, Kennedy was able to express an aggressive, unambiguous vision that to be realized, would eventually require the efforts of thousands of collaborating people building many complex systems with hundreds of thousands of interrelated components.

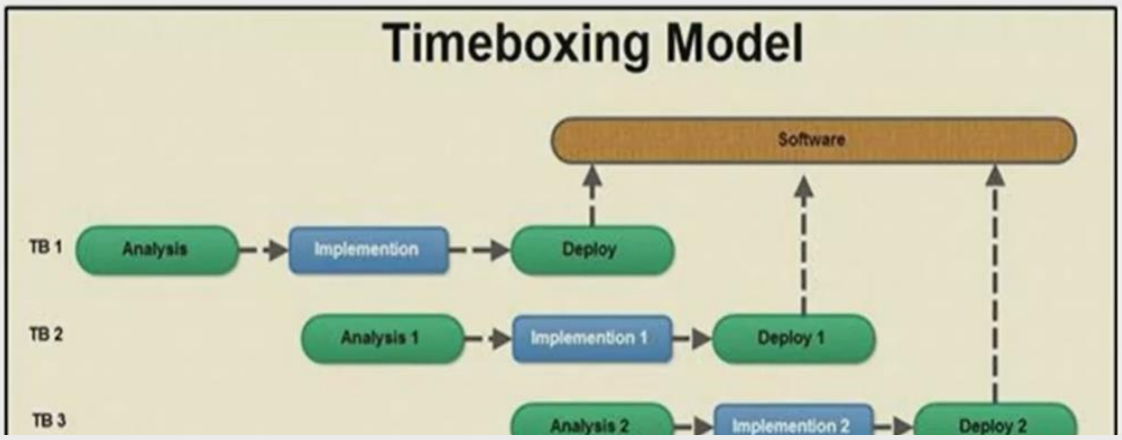
When developing products or services, many visions are expressed in terms of stakeholder value. This can be anything from achieving parity with competition or raising the parity bar, to targeting a new market, to

shortening the time to market, to redefining the game by changing market focus.

Popular vision formats, many of which are based on Jim Highsmith's 2009 Agile Product Management: Creating Innovative Products, include:

- **Elevator statement**: 30-second to 1-minute quick pitch of the product vision.
- **Product datasheet**: A 1-page marketing piece.
- **Product vision box**: Illustrate the box the product might ship in, including 3-4 points to emphasize on the label.
- **User conference slides**: 2-3 presentation slides that introduce the product at a user conference. Avoid bullet points.
- **Press release**: Write the ideal 1-page press release describing what is newsworthy about the new product when it becomes available.
- **Magazine review**: Draft a fictitious magazine review bylined by the solution reviewer in your industry's most popular trade magazine.

Timeboxing Model



Timeboxing Model

- In time boxing model, development is done iteratively as in the iterative enhancement model.
- In time boxing model, each iteration is done in a timebox of fixed duration.
- The functionality to be developed is adjusted to fit the duration of the timebox.
- Moreover, each timebox is divided into a sequence of fixed stages where each stage performs a clearly defined task (analysis, implementation, and deploy) that can be done independently.
- This model also requires that the time duration of each stage is approximately equal so that pipelining concept is employed to have the reduction in development time and product releases.

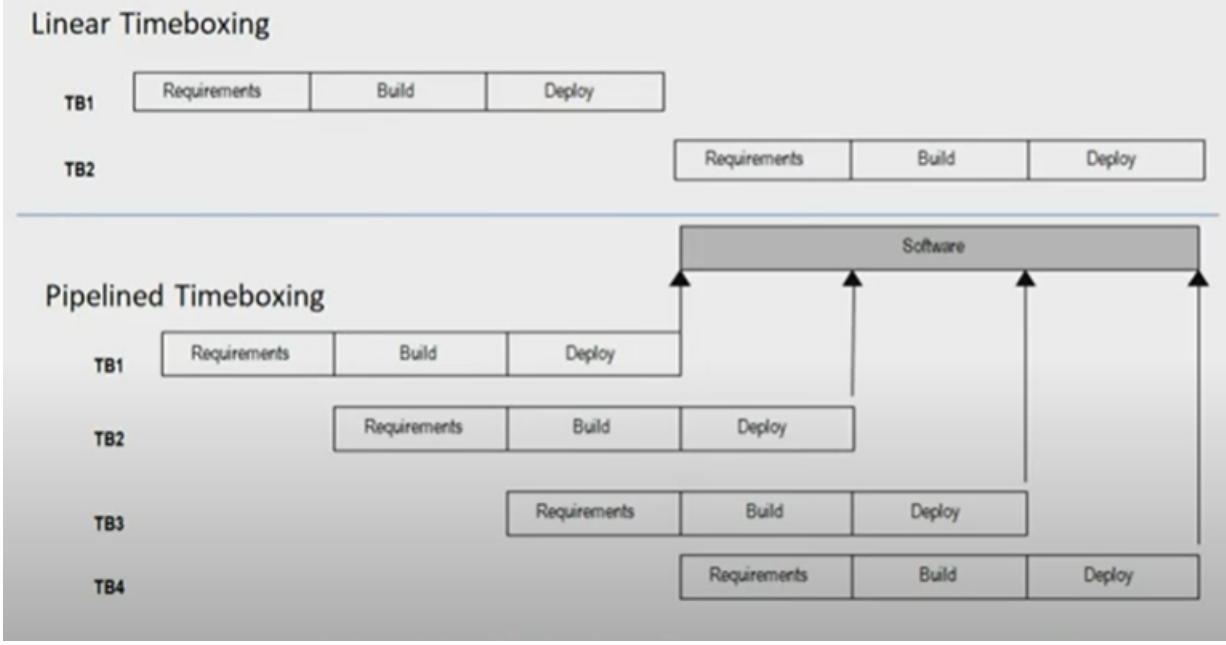
Timeboxing Model

- There is a individual team for each stage so that the work can be done in pipelining. Thus, stages should be chosen in such a way that each stage perform some logical unit of work that becomes the input for next stage.

Timeboxing

- Time boxing is an Iterative development but
  - fix an iteration duration, then determine the specifications
- Timeboxing model divide iteration in a few equal stages
- Use pipelining concepts to execute iterations in parallel

Timeboxing Execution



Time Boxed Iterations

- General iterative development – fix the functionality for each iteration, then plan and execute it
- In time boxed iterations – fix the duration of iteration and adjust the functionality to fit it
- Completion time is fixed, the functionality to be delivered is flexible

Linear Timeboxing

- This itself very useful in many situations
- Has predictable delivery times
- Overall product release and marketing can be better planned
- Makes time a non-negotiable parameter and helps focus attention on schedule
- Prevents requirements bloating
- Overall development time is still unchanged

Pipelined Timeboxing

- Multiple iterations executing in parallel
- Can reduce the average completion time by exploiting parallelism
- For parallel execution, can borrow pipelining concepts from hardware
- This leads to Pipelined Timeboxing Process Model

Pipelined Execution

- A Team starts executing it-1
- A Team finishes, hands over it-1 to B Team, starts executing it-2
- A Team finishes it-2, hands over to B Team; B Team finishes it-1, hands over to D Team; AT starts it-3, B Team starts it-2 (and D Team, it-1)