

every stage of software development to find out the defects and suggest improvements to produce good quality software.

4- Technical reviews

Software engineering is done in the form of modules. After completing each module, the software team assesses the completed module to find out and remove errors. It means that assessment of errors and corrections is done at each stage of activity. The team focuses on the quality of the software from the customer's point of view.

5- Measurement

This will include all the measurements of the project and product features i.e. quality of the software product. These measurements help the software team in delivering software that meets customers' needs.

6- Software configuration management

This activity is related to controlling and tracking changes in the software. The software team manages the effects of change throughout the software process.

7- Reusability management

In this activity, the software defines criteria for reuse of the work product (including software components). The quality of the software is good when the components of the software are developed for a certain application and are useful for developing other applications.

8- Work product preparation and production

LOC (line

10 (b)

stress delivery over analysis and design, and active and continuous communication between developers and customers.

3.1 AGILE AND AGILITY

The terms *agile* and *agility* are frequently used in the context of business transformations these days. Some people use agile and agility interchangeably as adjective and noun. Others treat them as different concepts. The initial definitions of *agile* and *agility* are as follows:

Agile

The word "agile" means the ability to move quickly and easily and also respond quickly to change. Agile is a way of thinking with a focus on collaboration, frequent delivery of value, and the ability to deal with functionality changes. It consists of values, principles, methods, and practices. Agile was invented in the software industry but nowadays businesses in all industry sectors make use of it.

In the software industry, the term *agile* is used to describe modern software development approaches that employ continual planning, learning, improvement, team collaboration, evolutionary development, and early delivery. It encourages flexible responses to change. Agile is a key aspect or feature of *agile software development*.

Agility

Agility is the ability of a business as a whole to respond quickly to changes, especially external changes. For example, by adapting business processes or changing customer experiences.

In software engineering, agility is the capability to manage various kinds of changes quickly during the software development process. It adopts the customer as a part of the development team. It also encourages team structures and attitudes that make communication more easily among the development team, and between the development team and the customer.

3.2 AGILE PROCESS MODEL

In traditional software development models like the *waterfall model*, a project can take several months or years to complete. These models (approaches) involve significant planning, designing, and documenting the system. The developers face various problems while using these models to develop software. For example, handling change requests and getting feedback from the customer during project development. A customer does not get to see the final software product until the end of the project.

To overcome the problems of developers, in the mid-1990s the *agile process model* was proposed. This model allowed the development team to focus on the software itself rather than on its design and documentation. The *agile model* universally relies on an incremental approach to software specification, development, and delivery. It is best suited to application development where the system requirements usually change rapidly during the development process.

The *agile model* is a combination of iterative and incremental process models. It focuses on process adaptability and customer satisfaction by rapid delivery of working software products. So, the main aim of the *agile model* is to complete the project quickly.

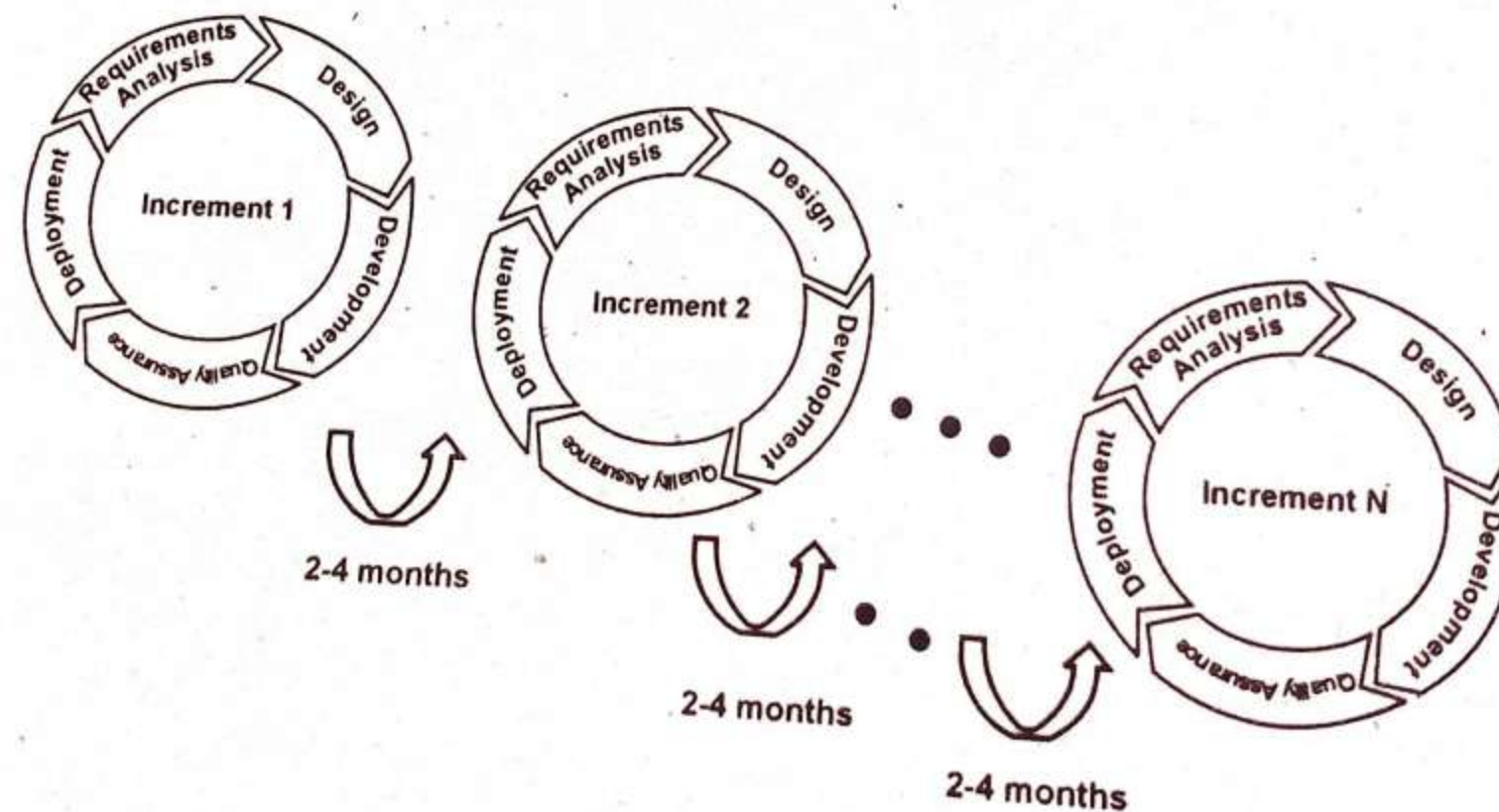


Figure 3.1: Agile Model

working
improvin
holds all
shown in

T
amount c
developn
to deliver
the custo

F
7

1-
I
c
i

2-
I
I
s

3-
I
,
(
I
I

4-
-

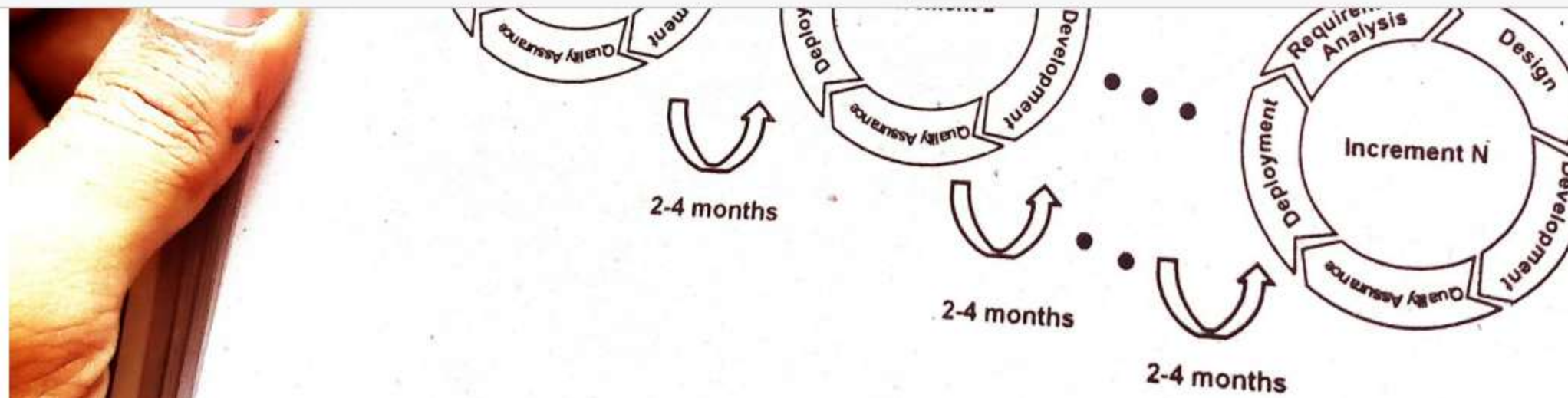


Figure 3.1: Agile Model

In the agile model, the tasks (or requirements) are divided into several smaller iterations or parts. The software is developed and released incrementally in iterations. It helps to minimize the project risk and to reduce the overall delivery time of the final software product. The project scope and requirements are decided at the beginning of the development process. Plans regarding the number of iterations, the duration, and the scope of each iteration are clearly defined in advance. Each iteration typically has the same time duration (between 2 to 4 weeks). Each

3-

4-

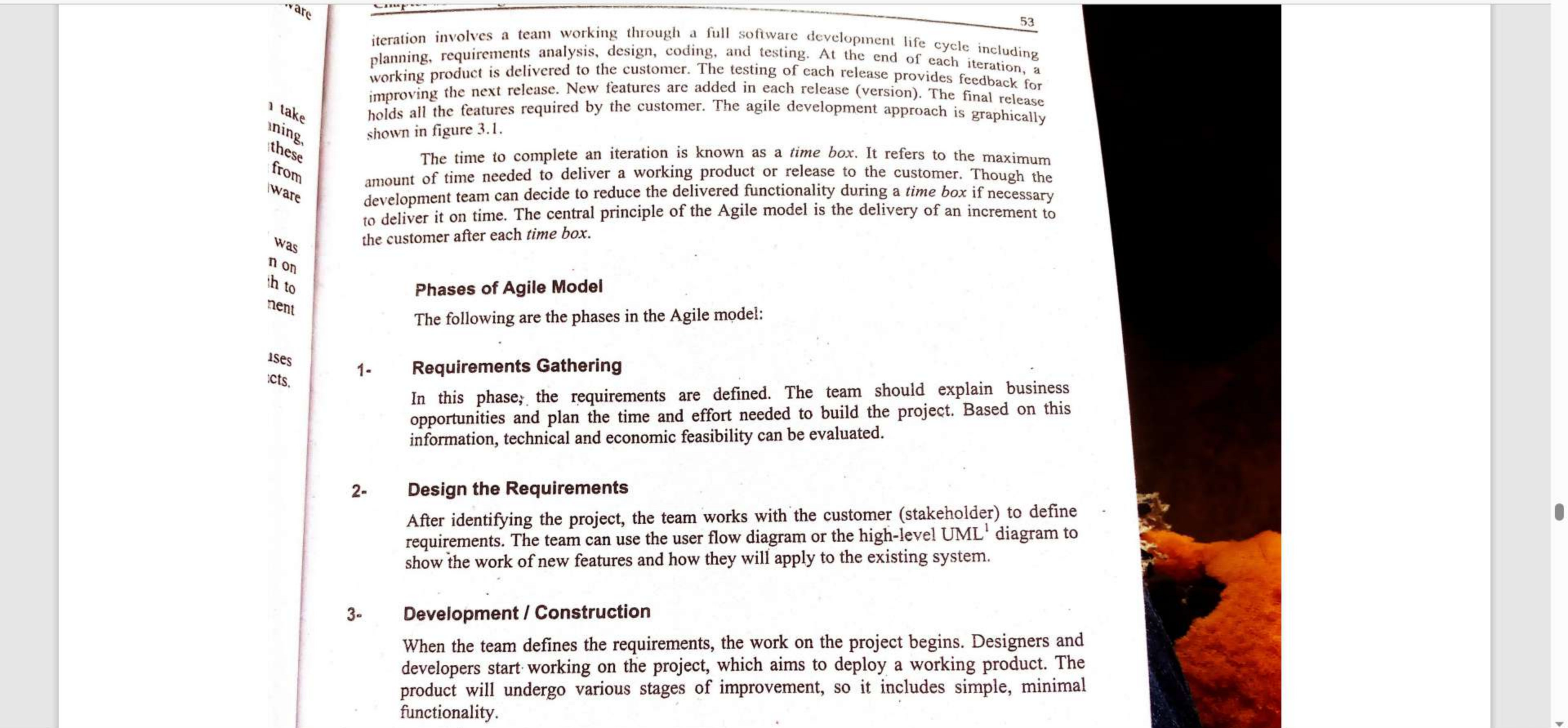
1 UML sta
both visua
language c
A UML m
the actual

Chapter #3 Agile Software Development

53

iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing. At the end of each iteration, a working product is delivered to the customer. The testing of each release provides feedback for improving the next release. New features are added in each release (version). The final release holds all the features required by the customer. The agile development approach is graphically shown in figure 3.1.

The time to complete an iteration is known as a *time box*. It refers to the maximum amount of time needed to deliver a working product or release to the customer. Though the development team can decide to reduce the delivered functionality during a *time box* if necessary to deliver it on time. The central principle of the Agile model is the delivery of an increment to the customer after each *time box*.



iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing. At the end of each iteration, a working product is delivered to the customer. The testing of each release provides feedback for improving the next release. New features are added in each release (version). The final release holds all the features required by the customer. The agile development approach is graphically shown in figure 3.1.

The time to complete an iteration is known as a *time box*. It refers to the maximum amount of time needed to deliver a working product or release to the customer. Though the development team can decide to reduce the delivered functionality during a *time box* if necessary to deliver it on time. The central principle of the Agile model is the delivery of an increment to the customer after each *time box*.

Phases of Agile Model

The following are the phases in the Agile model:

1- Requirements Gathering

In this phase, the requirements are defined. The team should explain business opportunities and plan the time and effort needed to build the project. Based on this information, technical and economic feasibility can be evaluated.

2- Design the Requirements

After identifying the project, the team works with the customer (stakeholder) to define requirements. The team can use the user flow diagram or the high-level UML¹ diagram to show the work of new features and how they will apply to the existing system.

3- Development / Construction

When the team defines the requirements, the work on the project begins. Designers and developers start working on the project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.

In this phase, the requirements are defined. The team should explain business opportunities and plan the time and effort needed to build the project. Based on this information, technical and economic feasibility can be evaluated.

2- Design the Requirements

After identifying the project, the team works with the customer (stakeholder) to define requirements. The team can use the user flow diagram or the high-level UML¹ diagram to show the work of new features and how they will apply to the existing system.

3- Development / Construction

When the team defines the requirements, the work on the project begins. Designers and developers start working on the project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.

4- Testing

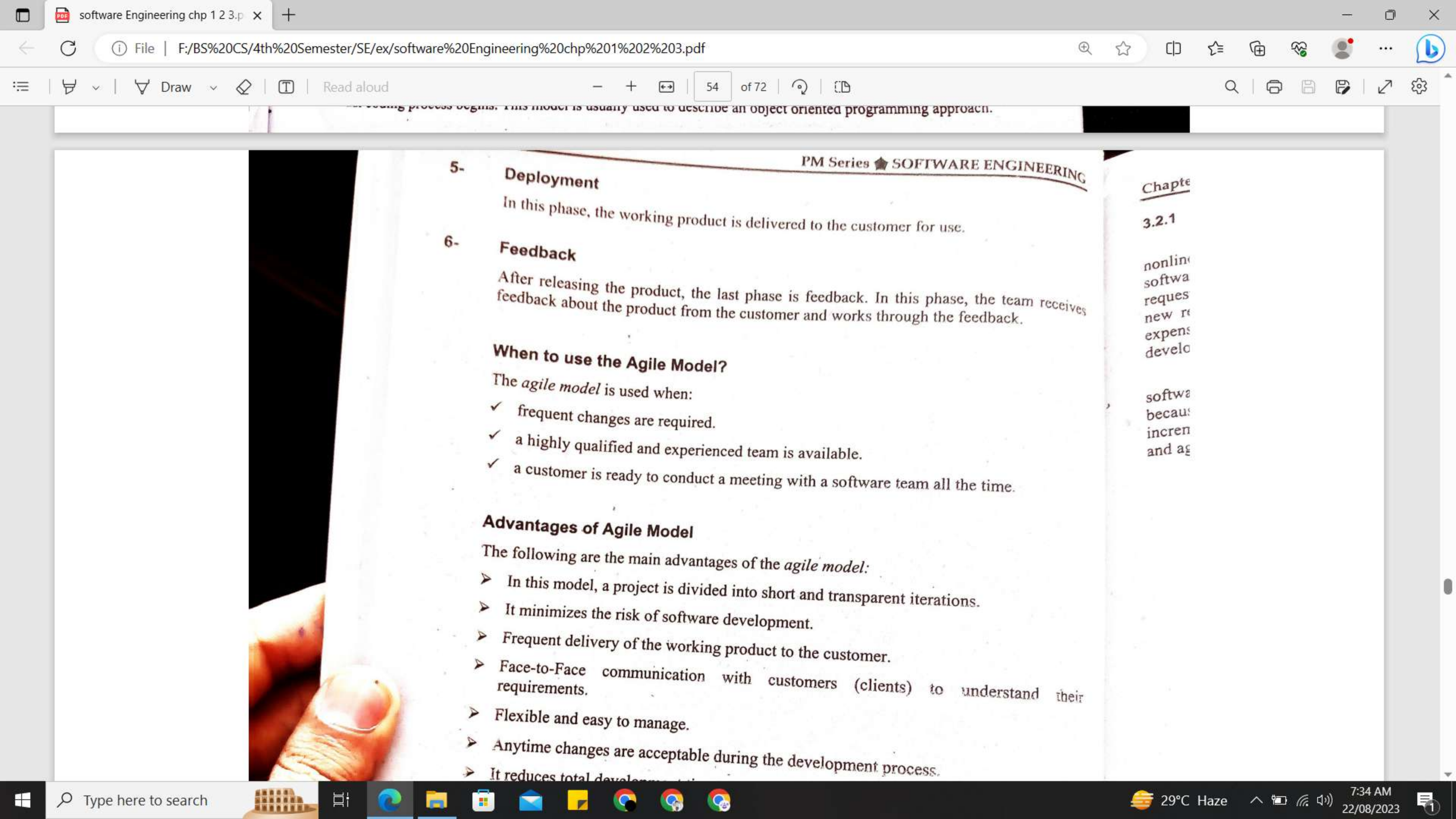
In this phase, the Quality Assurance team examines the product's performance and looks for the bugs (errors).

¹ UML stands for Unified Modeling Language. UML is a standard language which is used to describe a model, using both visual elements and text, just as in your map. Note that UML is not a programming language. It is a pictorial language consisting of symbols, diagrams, text, pseudo-code or anything that describes a software system.

A UML model or diagram is a representation of a software in terms of its structure, behavior and interactions, before the actual coding process begins. This model is usually used to describe an object oriented programming approach.

5- Deployment

In this phase, the working product is delivered to the customer for use.



5- Deployment

In this phase, the working product is delivered to the customer for use.

6- Feedback

After releasing the product, the last phase is feedback. In this phase, the team receives feedback about the product from the customer and works through the feedback.

When to use the Agile Model?

The *agile model* is used when:

- ✓ frequent changes are required.
- ✓ a highly qualified and experienced team is available.
- ✓ a customer is ready to conduct a meeting with a software team all the time.

Advantages of Agile Model

The following are the main advantages of the *agile model*:

- In this model, a project is divided into short and transparent iterations.
- It minimizes the risk of software development.
- Frequent delivery of the working product to the customer.
- Face-to-Face communication with customers (clients) to understand their requirements.
- Flexible and easy to manage.
- Anytime changes are acceptable during the development process.
- It reduces total development time.

Chapte

3.2.1

nonline
softwa
reques
new re
expens
develo

softwa
becaus
incren
and ag

a customer is ready to conduct a meeting with a software team all the time.

Advantages of Agile Model

The following are the main advantages of the *agile model*:

- In this model, a project is divided into short and transparent iterations.
- It minimizes the risk of software development.
- Frequent delivery of the working product to the customer.
- Face-to-Face communication with customers (clients) to understand their requirements.
- Flexible and easy to manage.
- Anytime changes are acceptable during the development process.
- It reduces total development time.

Disadvantages of Agile Model

The following are some disadvantages of the *agile model*:

- The development team should be highly professional and client-oriented.
- The shortage of formal documents creates confusion. The important decisions taken throughout various phases can be misinterpreted at any time by different team members.
- Due to the lack of proper documentation, once the project completes and the developers are allotted to another project, maintenance of the finished project can become a difficulty.

3.2.2

are as

1-

change is required for improvement in the quality of the software.

The above-mentioned five generic framework activities can be used during the development of simple applications, complex applications, Web applications, etc.

1.5.2 Umbrella Activities

Software engineering process framework also includes many *umbrella activities*. In general, umbrella activities are applied throughout a software project and help a software team to maintain the progress, quality, changes, and risks. Typical umbrella activities are as follows:

1- Software project tracking and control

In this activity, the software team accesses the project plan and compares it with the predefined schedule. If these project plans do not match with the predefined schedule, then the required actions are taken to maintain the schedule.

2- Risk management

In this activity, the software team assesses risks that may or may not occur. If the risks occur, then they can affect the outcome (result) of the project or the quality of the software product. Hence, proper risk management is required.

3- Software quality assurance (SQA)

In this activity, the software team defines and conducts the activities required to ensure software quality. For example, during software development, meetings are conducted at every stage of software development to find out the defects and suggest improvements to produce good quality software.

4- Technical reviews

Software engineering is done in the form of modules. After completing each module, the