BITS Pilani presentation

BITS Pilani
Pilani Campus

Dr. N. Jayakanthan

innovate achieve lead

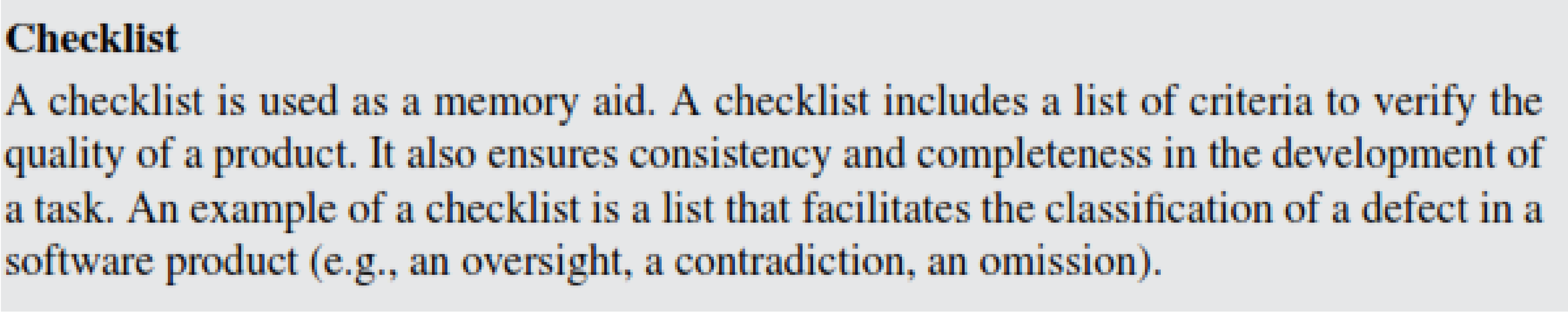
BITS Pilani
Pilani Campus

SE ZG501
Software Quality Assurance and Testing
Module – Session 16 Revision

Personal Review

Done by the person reviewing his own software product in order to find and fix the most defects possible.

- Principles of a personal review
  - Find and correct all defects in the software product.
  - Use a checklist produced from your personal data, if possible, using the type of defects that you are already aware of.
  - Follow a structured review process;
  - Use measures in your review;
  - Use data to improve your review;
  - Use data to determine where and why defects were introduced and then change your process to prevent similar defects in the future.



Checklist

A checklist is used as a memory aid. A checklist includes a list of criteria to verify the quality of a product. It also ensures consistency and completeness in the development of a task. An example of a checklist is a list that facilitates the classification of a defect in a software product (e.g., an oversight, a contradiction, an omission).

Practices - to develop an effective and efficient personal review

- Pause between the development of a software product and its review.
- Examine products in hard copy rather than electronically.
- Check each item on the checklist.
- Update the checklists periodically to adjust to your personal data.
- Build and use a different checklist for each software product.
- Verify complex or critical elements with an in depth analysis.

Desk-Check Reviews

- A type of peer review that is not described in standards is the desk-check review (Pass around) .

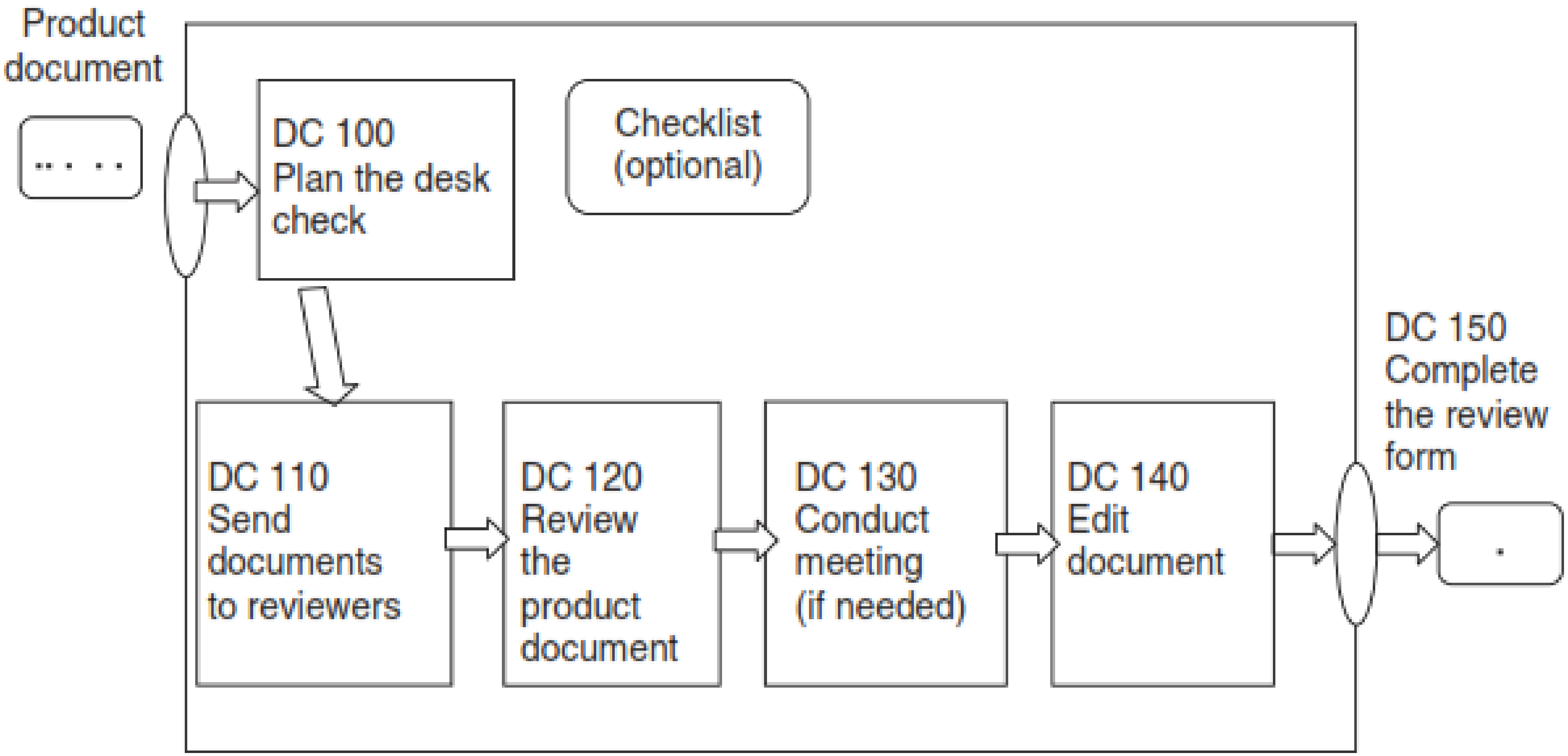

Product document
.. . . .
DC 100 Plan the desk check
Checklist (optional)
DC 110 Send documents to reviewers
DC 120 Review the product document
DC 130 Conduct meeting (if needed)
DC 140 Edit document
DC 150 Complete the review form
.

Figure 5.7 Desk-check review.

There are six steps.

- Initially, the author plans the review by identifying the reviewer(s) and a checklist.
- A checklist is an important element of a review as it enables the reviewer to focus on only one criterion at a time.
- A checklist is a reflection of the experience of the organization.
- Then, individuals review the software product document and note comments on the review form provided by the
- author.
- When completed, the review form can be used as “evidence” during an audit.

Important features of checklists:

- – Each checklist is designed for a specific type of document (e.g., project plan, specification document).
- – Each item of a checklist targets a single verification criteria.
- – Each item of a checklist is designed to detect major errors. Minor errors, such as misspellings, should not be part of a checklist.
- – Each checklist should not exceed one page, otherwise it will be more difficult to use by the reviewers.
- **– E**ach checklist should be updated to increase efficiency.
- – Each checklist includes a version number and a revision date.

Effective Test Management and Planning

- During the testing phase of software development, testing activities are managed well to complete the testing process smoothly and on time as well.
- **Test Management** is a process where testing activities are managed to ensure high-quality and high-end testing of software applications.
- This method consists of tracking, organization, controlling processes and checking the visibility of the testing process to deliver a high-quality software application.

- Test management is concerned with both test resource and test environment management.
- It is the role of test management to ensure that new or modified service products meet the business requirements for which they have been developed or enhanced.
- It makes sure the software testing process runs as expected.

key elements of Test Management

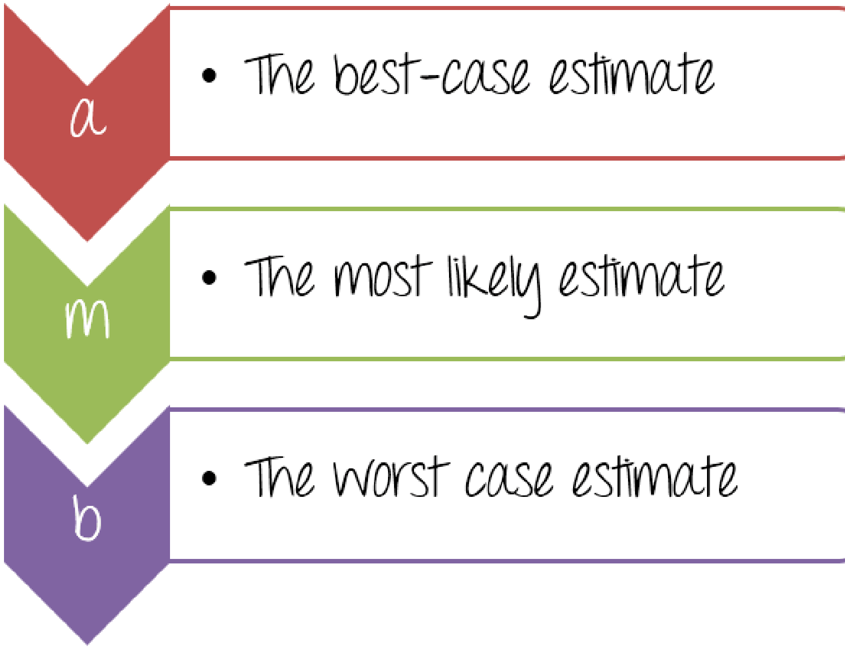
- Test organization
- Test planning
- Detailed test design and test specifications
- Test monitoring and assessment
- Product quality assurance

Test Estimation and Scheduling

- Test estimation techniques refer to the methods and approaches used to determine or estimate the effort, time, and resources required for testing activities in software development projects.
- Software test estimation is a managerial task that involves assessing and approximating the required time, resources, and costs for executing tests in a specific environment.
- It serves as a projection that aids in preventing time constraints and exceeding budgets.

Three Point Estimation

- Three-Point estimation is one of the techniques that could be used to estimate a task.
- The simplicity of the Three-point estimation makes it a very useful tool for a Project Manager that who wants to estimate.
- In three-point estimation, **three** values are produced initially for every task based on **prior experience** or **best-guesses** as follows:



a

- The best-case estimate

m

- The most likely estimate

b

- The worst case estimate

When estimating a task, the Test Manager needs to provide three values, as specified above.

The three values identified, estimate what happens in an **optimal state**, what is the **most likely**, or what we think it would be the **worst case scenario.**

Example: “**Create the test specification**”, for bank website

You can estimate as following

The **best case** to complete this task is **120** man-hours (around 15 days). In this case, you have a talented team, they can finish the task in smallest time.

The **most likely** case to complete this task is **170** man-hours (around 21 days). This is a normal case, you have enough resource and ability to complete the task

The **worst case** to complete this task is **200** man-hours (around 25 days). You need to perform much more work because your team members are not experienced.

Now, assign the value to each parameter as below

$$a = 120 \quad m = 170 \quad b = 200$$

The effort to complete the task can be calculated using **double-triangular distribution** formula as follows-

$$E = (a + 4m + b)/6$$

$$E = (120 + 4 * 170 + 200)/6$$

$$E = 166.6\ (man - hours)$$

- Parameter E is known as **Weighted Average**.
- It is the estimation of the task “Create the test specification”.

In the above estimation, you just determine a **possible** value, and not a **certain** one., we must know about the **probability** that the estimation is correct. You can use the other formula:

Since estimation involves uncertainty, it is essential to understand the **probability** that the estimated effort is correct

$$SD = (b - a)/6$$

$$SD = (200 - 120)/6$$

$$SD = 13.33\ (man - hours)$$

To measure this uncertainty, we use **Standard Deviation (SD)**, which quantifies how much the actual effort may **deviate from the estimated value**..

- Now you can conclude the estimation for the task “Create the test specification”
- To complete the task “Create the test specification” of Bank website, you need **166.6 ± 13.33** Man-hour (153.33 to 179.99 man-hour)

Test estimation best practices

Add some buffer time

Account Resource planning in estimation

Use the past experience as reference

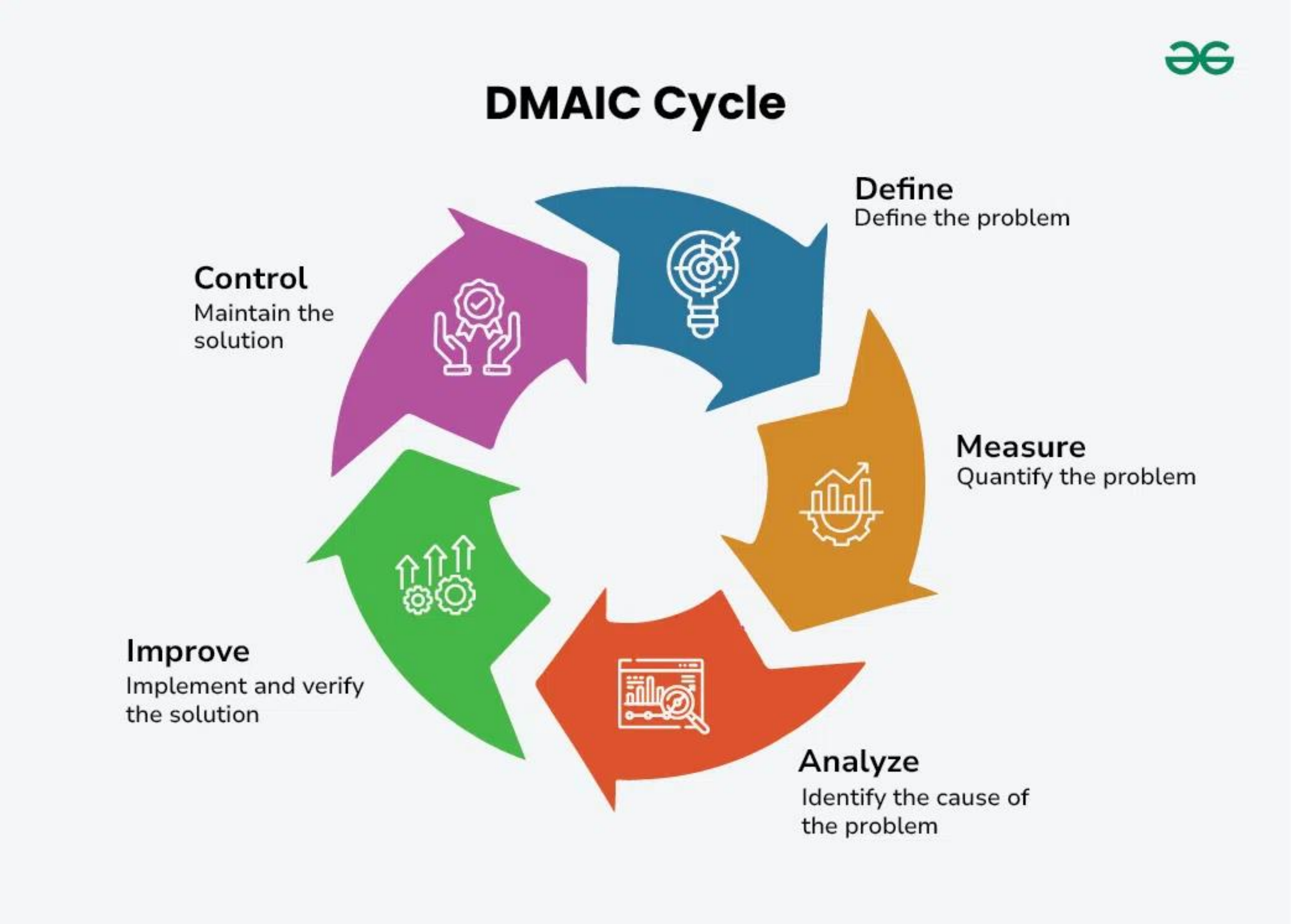
Stick to your estimation

DMAIC Six Sigma Methodology

DMAIC is used to **enhance an existing business process**.

A DMAIC project involves identifying important problem that are creating the problem, verifying those problem, brainstorming solutions, implementing them, and designing a control plan to maintain the improved state.

The DMAIC methodology is designed for the team who are responsible for improving a project.



DMAIC Cycle

Define
Define the problem

Measure
Quantify the problem

Analyze
Identify the cause of the problem

Improve
Implement and verify the solution

Control
Maintain the solution

The DMAIC project methodology has five phases:

- Define
- Measure
- Analyze
- Improve
- Control

Define

The Define phase of a DMAIC project involves identifying problems, establishing project requirements, and setting success goals.

Six Sigma leaders use specific tools during this phase to adapt the approach for different projects, based on leadership input, available resources, and budget.

Measure

In the **Measure** phase of DMAIC, the goal is to:

- **Collect data** to understand the current state of the process.
- **Use data to confirm or reject assumptions** about the problem.

This phase is important because reliable data is needed for effective analysis.

🔍 **Key Activities:**

- Gather and organize relevant data.
- Build tools or use software to extract information.
- Use filters or manual checks to process large data sets.
- Without good data, it's hard to measure or improve anything accurately.

Analyze

Analyze phase is a critical stage where the root causes of problems or inefficiencies within a process are identified and understood.

During the Analyze phase of a DMAIC project, teams develop predictions about relationships between inputs and outputs, use statistical analysis and data to validate the prediction and assumptions they've made thus far.

In a DMAIC project, the Analyses phase leads to the Improve phase, where hypothesis testing can confirm assumptions and potential solutions.

Improve

- Teams **develop and test solutions** based on what they learned during the **Analyze** phase.
- They use **statistics and real-life observations** to check if their ideas work.
- As they start applying the solutions, they keep doing **hypothesis testing** to make sure the changes are effective.

👉 The goal is to **fix the root cause** and make the process **better and more efficient**.

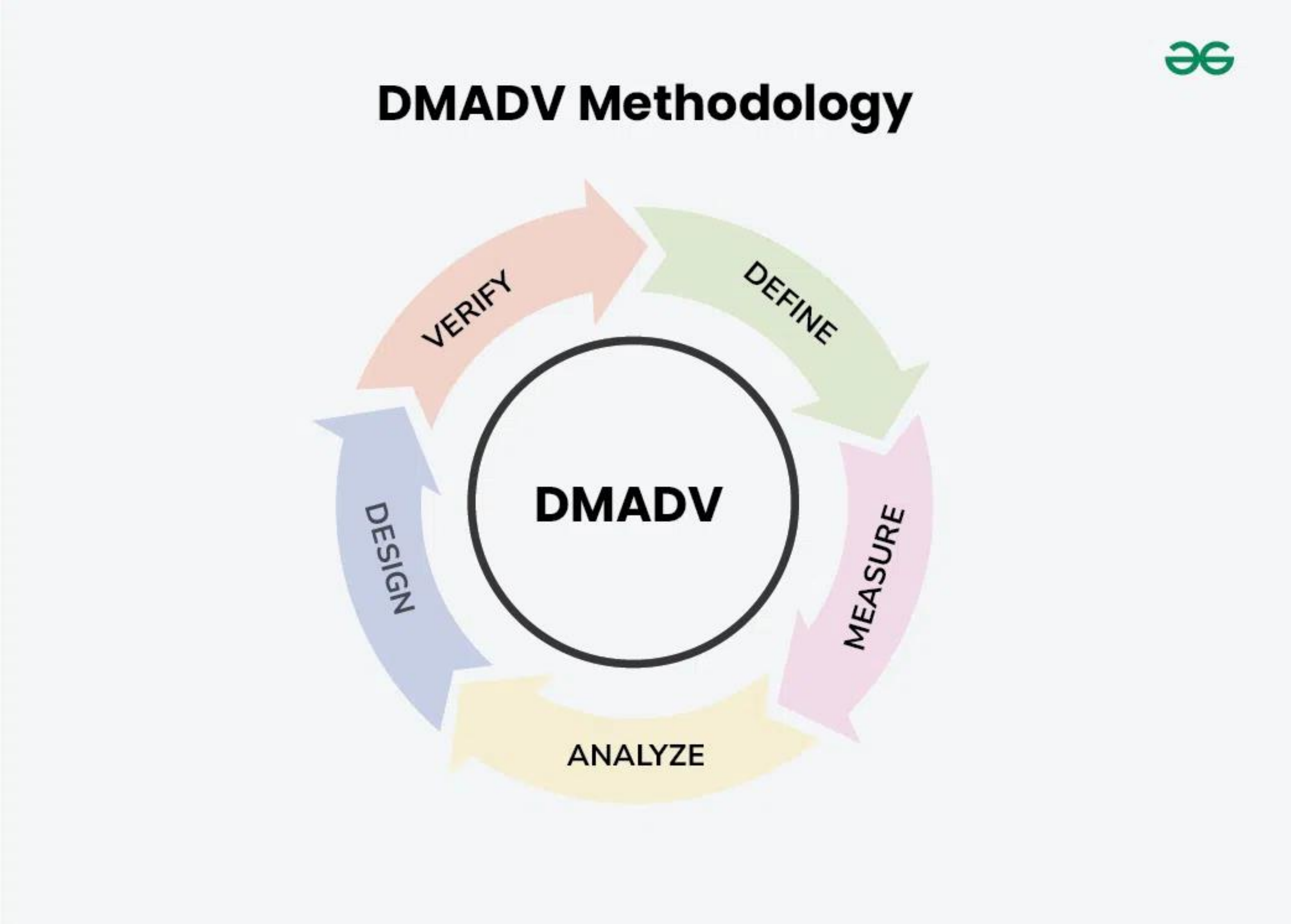
Control

- This is the **final step** of the DMAIC process.
- The goal is to **make sure the improvements last** over time.
- Teams create **rules, tools, and standards** to keep the process running smoothly.
- They also **train the people** (called **process owners**) who will manage the improved process every day.

DMADV Six Sigma Methodology

DMADV is used to create new product designs or process designs. Six Sigma teams use DMADV in the following scenario:

- The organization wants to launch a new service or product.
- Business leaders decide to replace a process to meet upgrade requirements or to align business processes, machinery, or workers with future goals.
- A Six Sigma team learns that **upgrading an existing process is unlikely to achieve the expected outcomes**, and a new design is necessary to meet quality and performance standards..


DMADV Methodology
DEFINE
MEASURE
ANALYZE
DESIGN
VERIFY
DMADV

The DMADV project methodology also has **five** phases:

- Define
- Measure
- Analyze
- Design
- Verify

Define

In the **Define** stage of a DMADV project, the focus is on understanding the problem clearly and setting project goals.

- Teams **identify the problem** or opportunity.
- Requirements are defined **based on customer needs** and expectations.
- If a **change management program** is already in place in the organization, its needs must be integrated into this stage.
- This phase sets the **foundation** for all future design work, making it more structured and clear than in DMAIC.
- Define stage is slightly more strict than in DMAIC.

Measure

During the DMADV Measure phase, teams use data to validate assumptions about the process and problem. Validation of assumptions also makes it into the analysis step. **The measurement phase focuses on collecting and arranging data for analysis**.

Analyze

Analyze phase is a critical stage where the root causes of problems or inefficiencies within a process are identified and understood. They priorities identifying best practices and standards for measuring and designing new processes.

Design

The **Design** phase is where DMADV begins to diverge significantly from DMAIC.

- In this stage, teams **create a detailed blueprint** of the new process or product.
- The focus is on building a solution from the ground up, guided by insights gained in earlier phases.
- The design process includes:
  - **Solution testing and prototyping**
  - **Process mapping and workflow planning**
  - **Infrastructure and system development**
  - **User experience and operational efficiency considerations**

👉 **Key Focus:** Designing a robust, scalable solution that meets customer and business needs.

Verify Phase

The Verify phase in DMADV checks if the designed solutions work as intended, measuring their success against initial goals, ensuring improvements are effective and sustainable.

Difference between DMAIC and DMADV

The primary difference between **DMAIC** and **DMADV** lies in their **team goals** and **project outcomes**:

- **DMAIC** is used to **improve existing processes**.
- **DMADV** is used to **design new processes or products** from the ground up.

Both methodologies aim to:

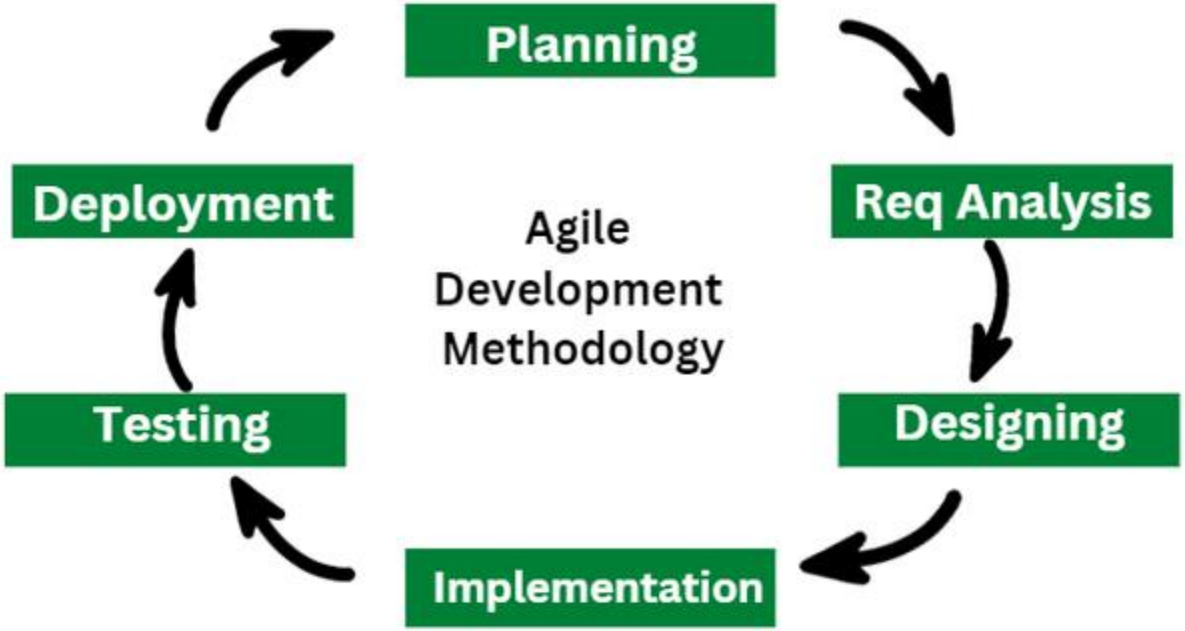
- Deliver **better quality**
- Improve **efficiency and productivity**
- Increase **profits**
- Ensure **customer satisfaction**

Introduction to Agile Methodology and Testing

Agile is a project_management and software development approach that aims to be more effective.

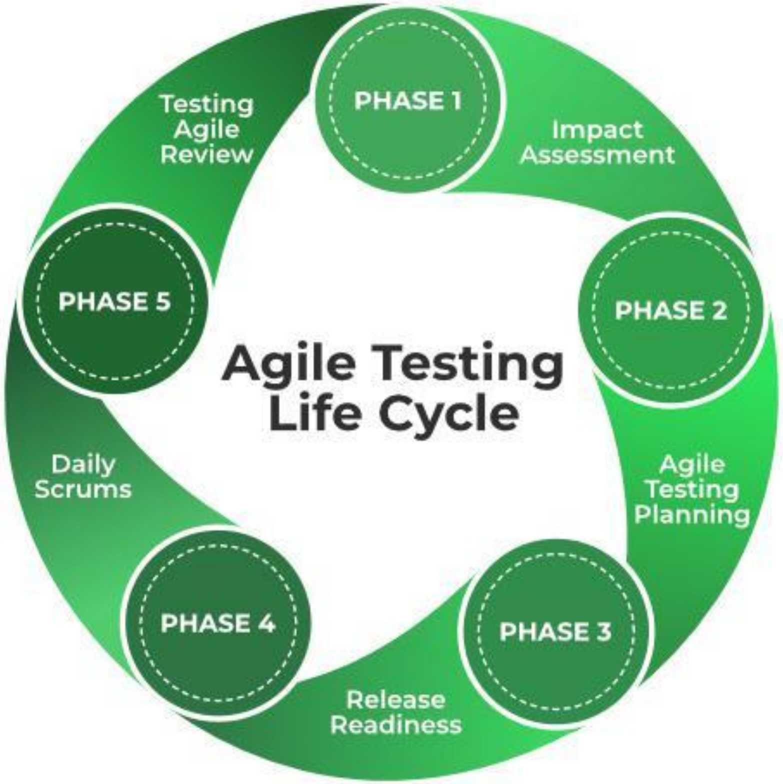
- It focuses on delivering smaller pieces of work regularly instead of one big launch.
- This allows teams to adapt to changes quickly and provide customer value faster.

Life cycle of Agile Methodology



- In Agile, testing is an ongoing activity rather than a distinct phase, focusing on collaboration between testers, developers, and stakeholders.
- This approach helps detect and fix issues early, aligning with Agile's iterative development cycle.

Agile Testing Life Cycle



Agile Testing Life Cycle :
5 different phases:

Impact Assessment: This is the first phase of the agile testing life cycle also known as the feedback phase where the inputs and responses are collected from the users and stakeholders. This phase supports the test engineers to set the objective for the next phase in the cycle.

Agile Testing Planning: In this phase, the developers, customers, test engineers, and stakeholders team up to plan the testing process schedules, regular meetings, and deliverables.

Release Readiness: This is the third phase in the agile testing lifecycle where the test engineers review the features which have been created entirely and test if the features are ready to go live or not and the features that need to be sent again to the previous development phase.

Daily Scrums: This phase involves the daily morning meetings to check on testing and determine the objectives for the day. The goals are set daily to enable test engineers to understand the status of testing.

Test Agility Review: This is the last phase of the agile testing lifecycle that includes weekly meetings with the stakeholders to evaluate and assess the progress against the goals.

Agile Test Plan

1. **Purpose**:
   1. In Agile development, a test plan is essential for every iteration or release.
2. **What It Includes**:
   1. **Types of Testing**: Specifies the kinds of tests (e.g., unit, integration, regression) that will be conducted.
   2. **Test Data Requirements**: Defines what test data is needed.
   3. **Test Environments**: Describes where the tests will be executed (e.g., staging servers, test labs).
   4. **Test Results**: Captures and evaluates the outcomes of the testing activities.
3. **Frequency**:
   1. A new **test plan is created and updated for every release** to reflect changes in features or code.

The Agile test plan includes the following:

- Test Scope.
- Testing instruments.
- Data and settings are to be used for the test.
- Approaches and strategies used to test.
- Skills required to test.
- New functionalities are being tested.
- Levels or Types of testing based on the complexity of the features.
- Resourcing.
- Deliverables and Milestones.
- Infrastructure Consideration.
- Load or Performance Testing.
- Mitigation or Risks Plan.