



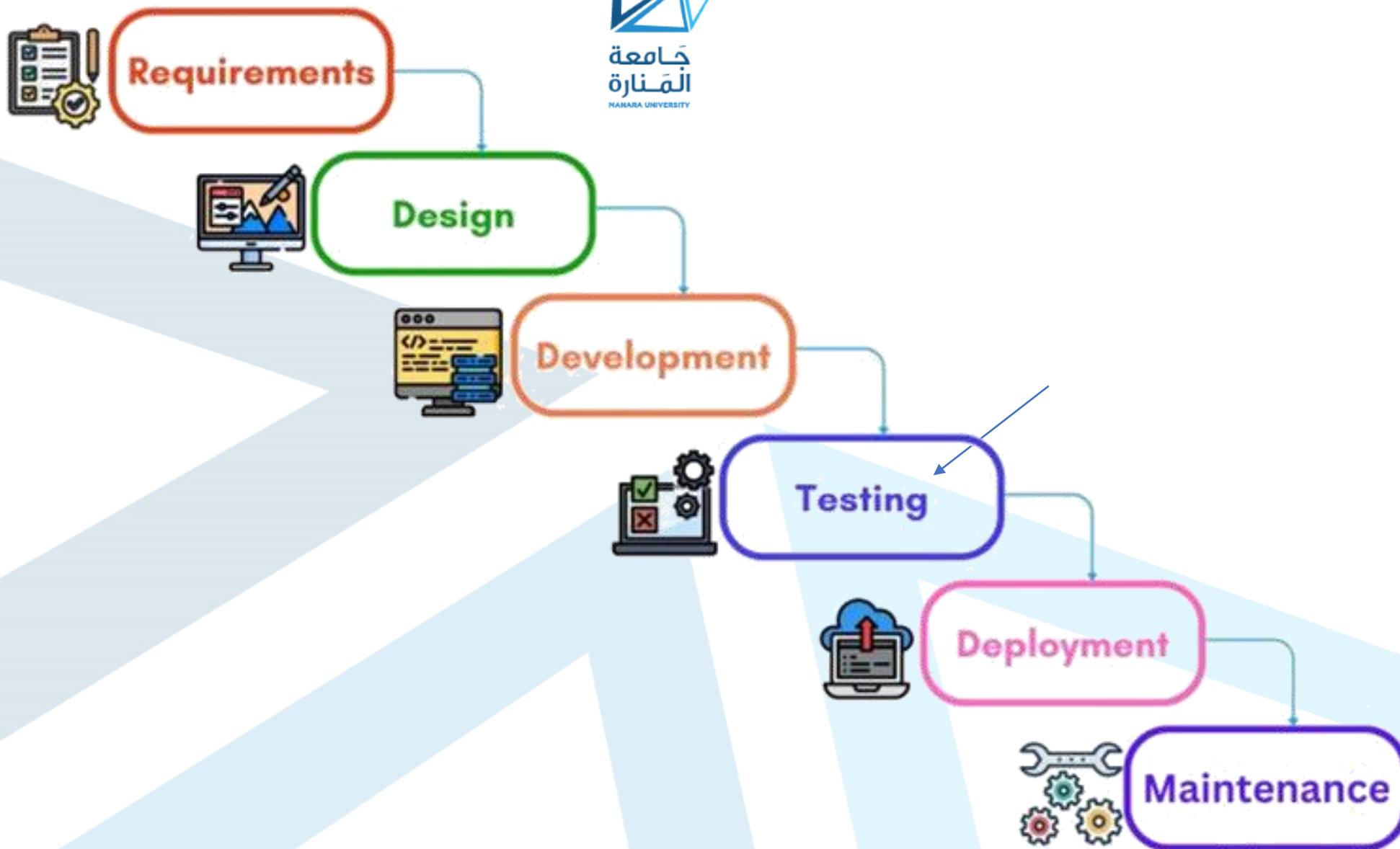
Software Engineering -2-

Lecture -4-

Dr. Inas Laila

الفصل الدراسي الاول ٢٠٢٥ / ٢٠٢٦





What Is Software Testing

تُعد مرحلة اختبار البرمجيات في دورة حياة تطوير البرمجيات (SDLC) مرحلةً حاسمةً لضمان الجودة، تحدث بعد مرحلة التطوير وقبل النشر، وتُعدّ بمثابة البوابة الرئيسية لجودة البرمجيات.

عادةً ما يهدف اختبار البرمجيات إلى تشغيل برنامج أو تطبيق بهدف اكتشاف ثغرات برمجية (أخطاء أو عيوب أخرى). ويعني اختبار البرمجيات تشغيلها في ظروف مُتحكم بها، لـ

- Verify that it behaves “as specified” ,
- To detect Defects / Errors ,
- To validate that what has been specified is what the user actually wanted.

إن مهمة الاختبار هي عملية تكرارية، حيث عندما يتم إصلاح خطأ واحد، فإنه يمكن أن يؤدي إلى تسليط الضوء على أخطاء أخرى أعمق، أو حتى إنشاء أخطاء جديدة.

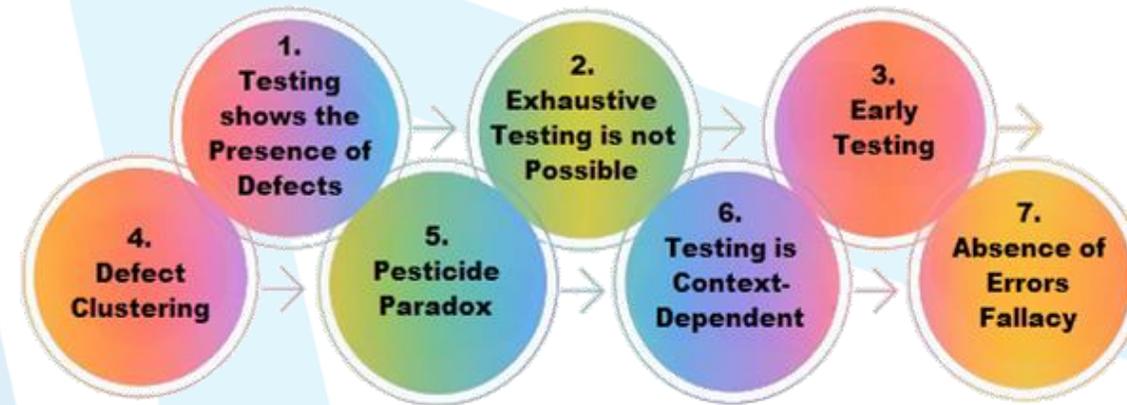


The 7 Quality Principles and Concepts

1. Testing shows presence of Defects
2. Exhaustive Testing is Impossible! (الاختبار الشامل مستحيل)
3. Early Testing
4. Defect Clustering
5. The Pesticide Paradox
6. Testing is Context Dependent (الاختبار يعتمد على السياق)
7. Absence of Errors Fallacy (غياب الأخطاء مغالطة)

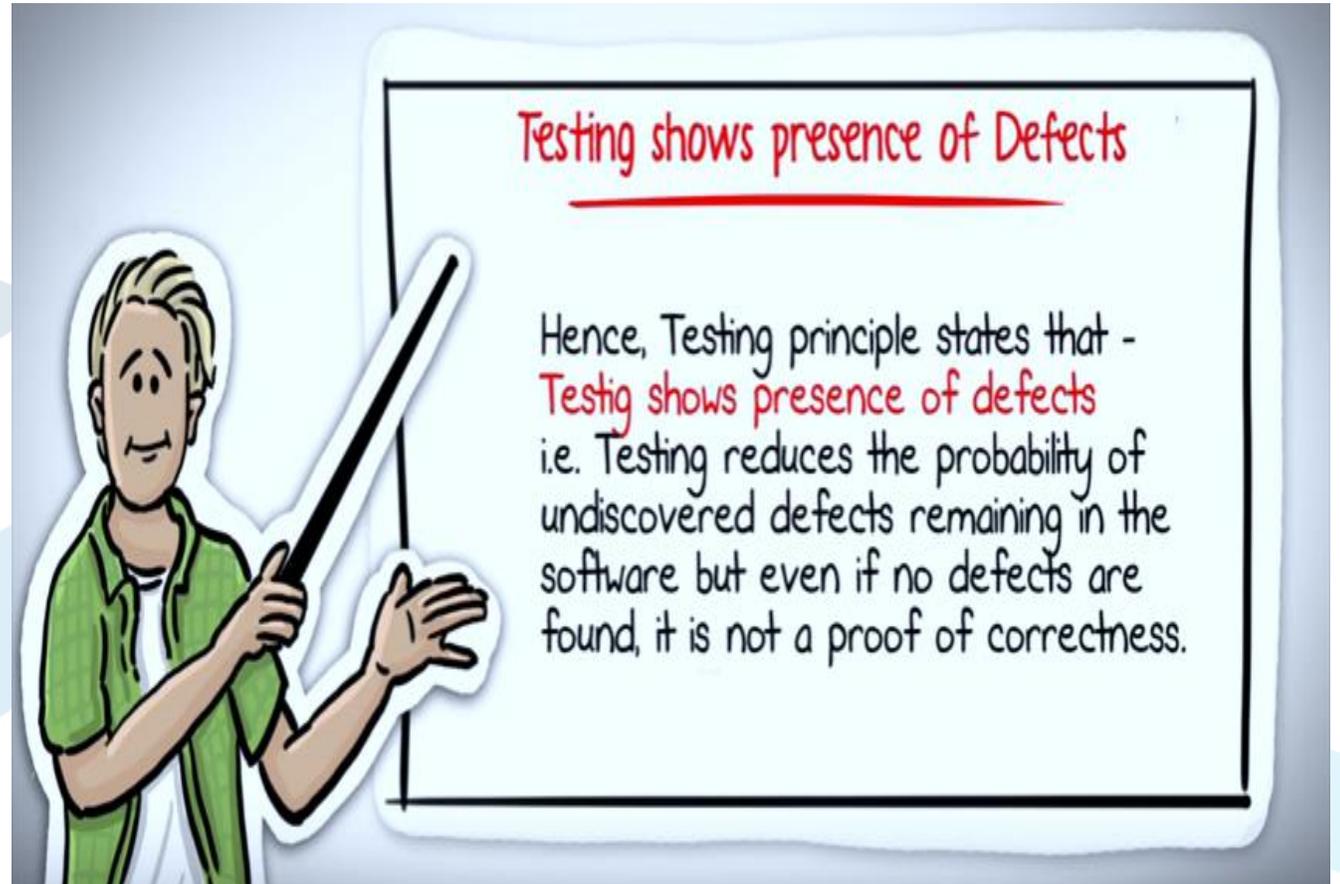
من خلال تطبيق هذه المبادئ، يوفر المختبرون الوقت، ويقللون التكاليف، ويقدمون تطبيقات عالية الجودة تتوافق مع أهداف العمل.

7 Principles of Software Testing



1- Testing shows the presence of Defects

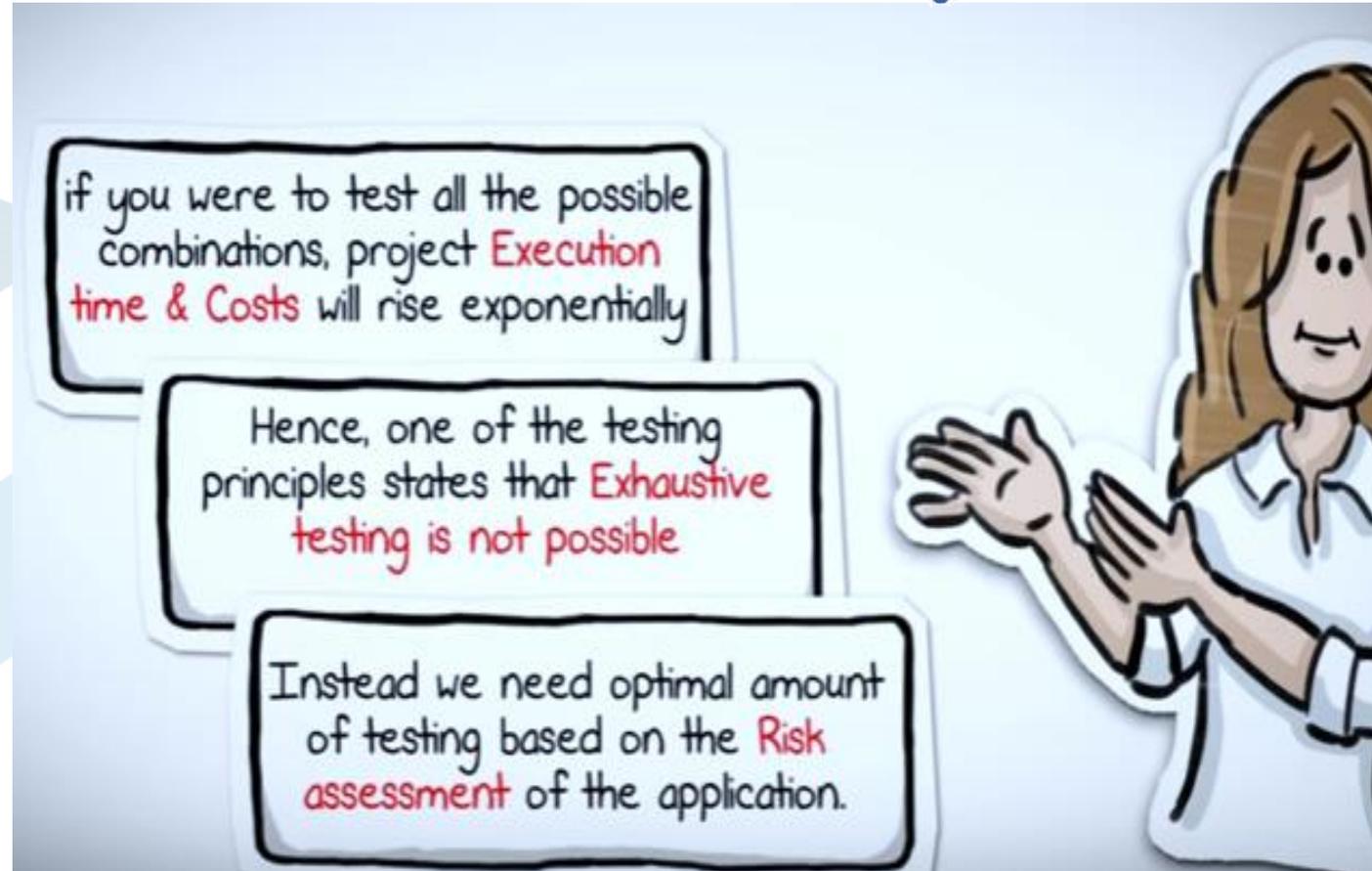
We test to find Faults (a.k.a Defects). As we find more defects, the probability of undiscovered defects remaining in a system reduces. However Testing cannot prove that there are no defects present



2- Exhaustive Testing is Impossible!

We have learned that we cannot test everything (i.e. it is not possible to test all the functionalities with all valid and invalid combinations of input data during actual testing). That is we must Prioritise our testing effort using a Risk Based Approach. Exhaustive Testing is impractical and impossible?

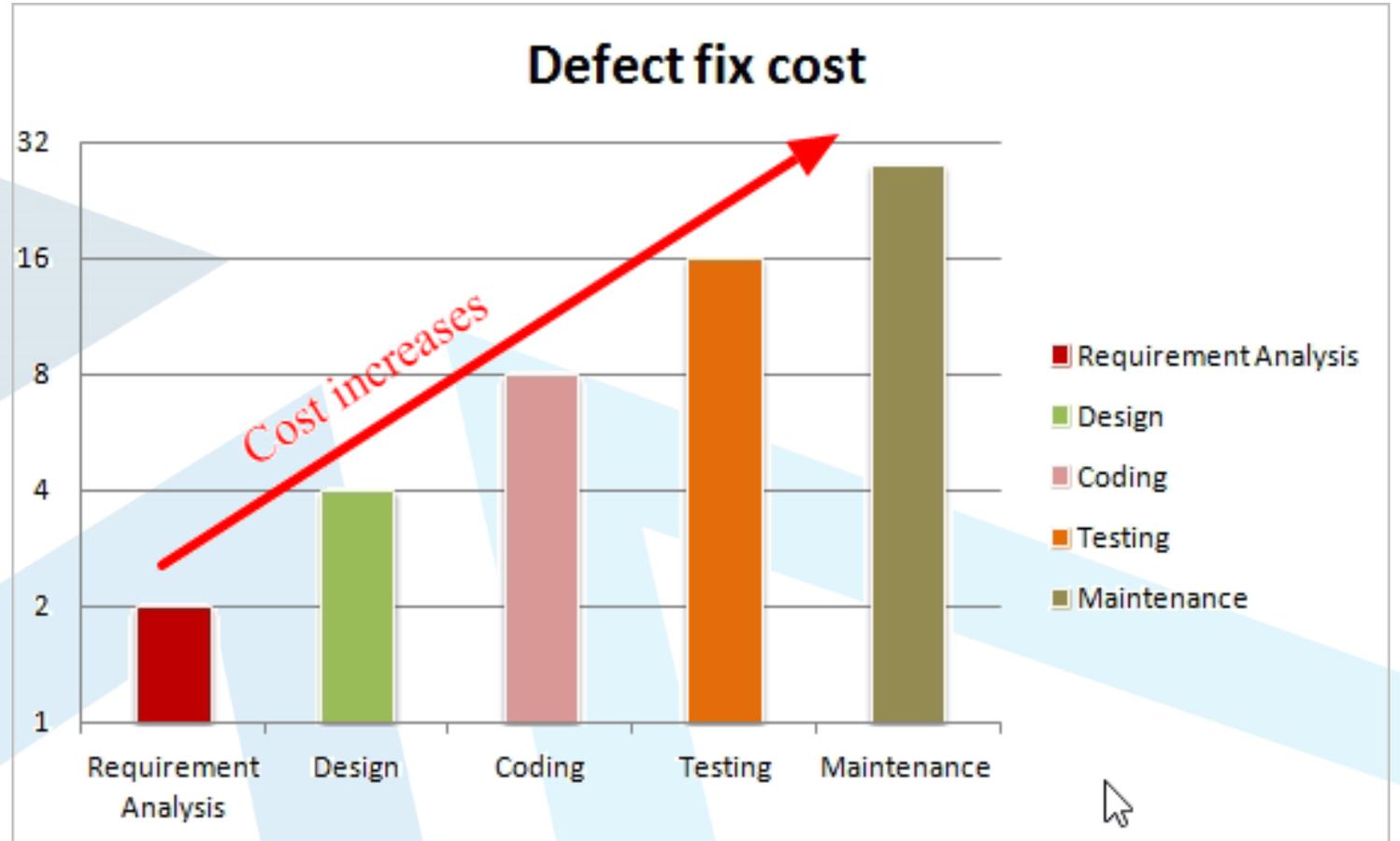
Hence it is recommended to test input data using different methods like Equivalence Partitioning and Boundary Value Analysis.



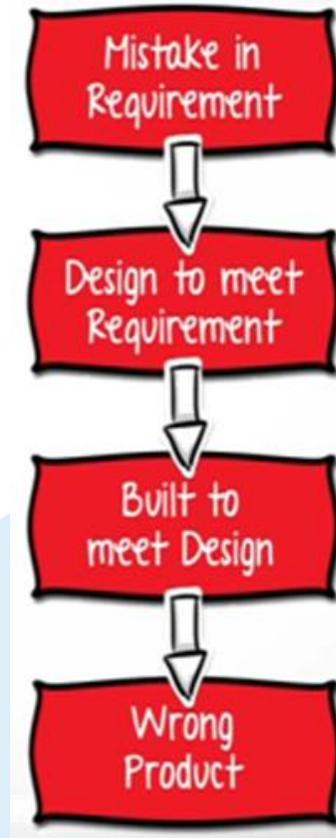
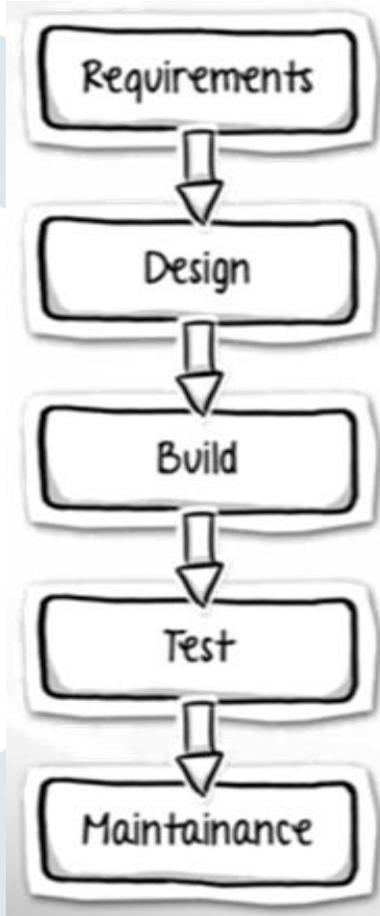
3- Early testing

Testing activities should start as early as possible in the development life cycle

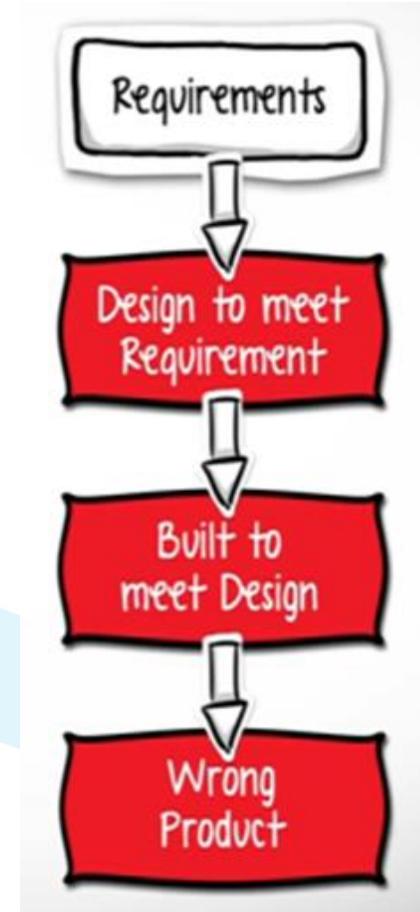
The earlier in life cycle a defect is detected, the cheaper it is to fix it.



WATERFALL MODEL



You will have start afresh with project



Re-Design to correct defects



SDLC V-Model

The V-Model stands as one of the most systematic and **quality-focused** Software Development Life Cycle (SDLC) methodologies. It is also known as the Verification and Validation model.

It extends **the traditional waterfall model** by **emphasizing verification and validation** at each phase of development.

This model focuses on verification and validation activities early in the life cycle thereby enhancing the probability of building an error-free and good quality product



Validation and Verification process

What is Verification in Software Testing? The main goal of verification process is to ensure quality of software application, design, architecture etc in order to check if the software has been built according to the requirements or not.. essentially answering the question "Are we building the product right?" **Verification finds the bugs early in the development cycle**

What is Validation in Software Testing? **Validation in Software Engineering** is **a dynamic** mechanism of testing and validating if the software product actually **meets the exact needs of the customer or not.**

- Verification process comes before validation whereas Validation process comes after verification.
- Verification process includes checking of documents, design, code and program whereas Validation process **includes testing and validation of the actual product** addressing the question **"Are we building the right product?"**.
- **Verification does not involve code execution while Validation involves code execution.**
- **Verification uses methods like reviews, walkthroughs and static analysis whereas Validation uses methods like black box testing, white box testing**



The V-Model consists of two main phases:

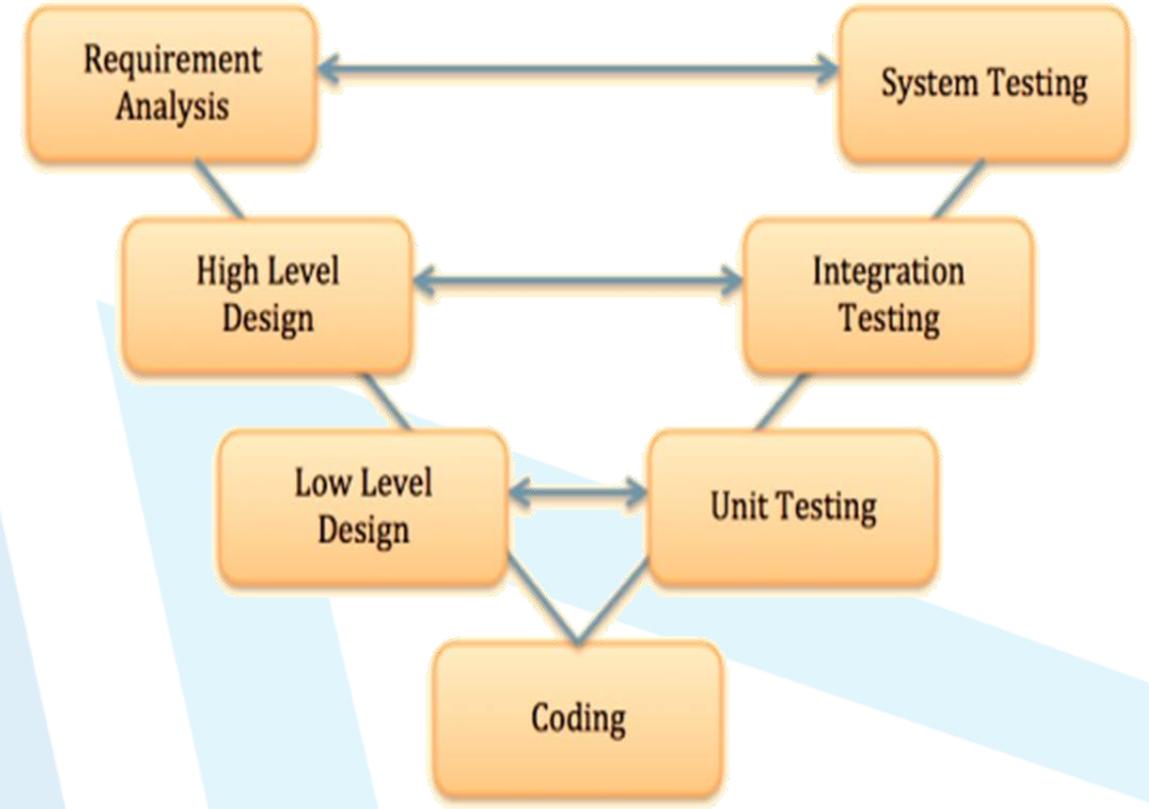
Verification Phase of V-Model (left side of V)

Validation Phase of V-Model (right side of V)

يتكون النموذج V من مرحلتين رئيسيتين: مرحلة **Verification** وهي الجانب الأيسر من النموذج (V) وهي تركز على الاختبار في مرحلتي تحليل وتصميم النظام قبل بدء البرمجة. أما مرحلة **Validation** فهنا يتم الاختبار بعد عملية البرمجة.

مزايا النموذج V

يشجع على الكشف المبكر عن العيوب، مما يقلل التكلفة وإعادة العمل.
يضمن جودة عالية للمخرجات من خلال عمليات تحقق دقيقة.



V Model



يقدم الجدول مقارنة بين نموذج الشلال وV-Model

النموذج	مراحل واضحة؟	مرونة التغيير	علاقة الاختبار
الشلال	نعم	منخفضة	بعد التطوير
V-Model	نعم	منخفضة	مع كل مرحلة

SDLC: SDLC is Software Development Life Cycle: It is the sequence of activities carried out by Developers to design and develop high-quality software.

STLC: STLC is Software Testing Life Cycle: is a sequence of specific activities conducted during the testing process to **ensure software quality goals are met.**

Software Testing is not just a single/isolate activity, i.e. testing. It consists of a series of activities carried out methodologically. STLC involves both verification and validation activities.



4. Defect Clustering

في معظم الحالات، يمكن ملاحظة أن أغلب العيوب المكتشفة تكون موجودة في عدد صغير من الوحدات البرمجية modules، أي أن العيوب غالباً لا تكون موزعة عبر التطبيق بل تتكثف وتتجمع في اجزاء معينة من التطبيق. يعتمد هذا القياس على مبدأ باريتو



During testing, it may happen that most of the defects found are related to a small number of modules. There might be multiple reasons for this like the modules may be complex, coding related to such modules may be complicated etc.

This is the Pareto Principle of software testing where 80% of the problems are found in 20% of the modules.



5- The Pesticide Paradox

Pesticide Paradox principle says that if the same set of test cases are executed again and again over the period of time then these set of tests are not capable enough to identify **new defects in the system**. If required a new set of test cases can be added and the existing test cases can be deleted if they are not able to find any more defects from the system.

- ✓ If the same tests are repeated over and over again, eventually the same test cases will no longer find new bugs
- ✓ This is the another principle of testing called "Pesticide paradox"
- ✓ To overcome this, the test cases needed to be regularly reviewed & revised, adding new and different test cases to help find more defects.



مفارقة المبيدات: يجب علينا إنشاء حالات اختبار جديدة لاكتشاف أخطاء جديدة.



6- Testing is Context Dependent

There are several domains available in the market like Banking, Insurance, Medical, Travel, Advertisement etc and each domain has a number of applications. Also for each domain, their applications have different requirements, functions, different testing purpose, risk, techniques etc. Different domains are tested differently, thus testing is purely based on the context of the domain or application.

يتم اختبار المجالات المختلفة بشكل مختلف، وبالتالي يعتمد الاختبار بشكل كامل على سياق المجال أو التطبيق. على سبيل المثال، يتم اختبار البرامج ذات الأهمية الحرجة للسلامة بشكل مختلف عن برامج الألعاب. كما ان أن نسبة التكلفة المحددة للاختبار من اجمالي تكاليف تطوير البرمجة تختلف حسب التطبيق ودرجة المخاطر فيه

What is (KLOC): KLOC (thousands of lines of code) is a traditional measure of how large a computer program is



7- Absence of Errors Fallacy

إذا تم اختبار البرنامج بالكامل ولم يتم العثور على أي عيوب قبل الإصدار، فيمكننا القول أن البرنامج خالٍ من العيوب بنسبة ٩٩٪. ولكن ماذا لو تم اختبار هذا البرنامج وفقاً لمتطلبات خاطئة؟ في مثل هذه الحالات، حتى العثور على العيوب وإصلاحها في الوقت المناسب لن يساعد، حيث يتم إجراء الاختبار بناءً على متطلبات خاطئة لا تتوافق مع احتياجات المستخدم النهائي. حيث أن المتطلبات قد تم تفسيرها بشكل خاطئ وهنا حتى اكتشاف المزيد من العيوب لا يساعد في نقل التطبيق إلى المرحلة التالية أو إلى بيئة الإنتاج.

إذا كان برنامجك أو نظامك لا يلبي رغبات المستخدمين واحتياجاتهم وتوقعاتهم فلا يهم عدد العيوب التي تم اكتشافها وإصلاحها، فهو لا يزال غير صالح للاستخدام. أي ان اكتشاف العيوب وإصلاحها لا يُجدي نفعًا إذا كان النظام المنشأ لا يُلبي احتياجات المستخدمين وتوقعاتهم.



quality software:

- 1) The producer's view of quality software means meeting requirements.
- 2) Customer's/User's of software view of quality software means fit for use.



The 2 software Quality Gaps:

الفجوة الأولى هي فجوة المنتج: وهي الفجوة بين ما يتم تسليمه فعليًا وبين المتطلبات الموصفة.

الفجوة الثانية هي فجوة المستهلك او العميل: وهي الفجوة بين ما يتم تسليمه فعليًا وبين ما يتوقعه العميل.



To close the **Customer's gap** we need :

- **Customer surveys:**

يمكن تعريف استطلاعات العملاء بأنها طريقة لطلب تعليقات العملاء وملاحظاتهم حتى تتمكن المؤسسات من قياس رضا العملاء وفهم توقعاتهم

- **JAD (joint application development) sessions:**

جلسات التطوير المشترك للتطبيقات هي منهجية تعمل على تضمين العميل أو المستخدم النهائي في تصميم وتطوير التطبيقات من خلال سلسلة من ورش العمل التعاونية المعروفة باسم جلسات JAD وهي ورش عمل منظمة تجمع أصحاب المصلحة والمستخدمين النهائيين والمطورين ومحلي الأعمال وغيرهم من المشاركين الرئيسيين لتحديد متطلبات البرمجيات وتصميم الحلول **واتخاذ القرارات الحاسمة للمشروع بشكل تعاوني.**

صُممت هذه الجلسات لتسريع عملية التطوير من خلال إشراك جميع الأطراف المعنية منذ البداية، مما يضمن فهمًا مشتركًا لأهداف المشروع ويقلل من احتمالية سوء الفهم أو التعديلات المكلفة لاحقًا في دورة حياة المشروع.



سيكولوجيا الاختبار Psychology of Testing

العلاقة بين المطور والمختبر ليست عادةً علاقة سهلة للأسباب التالية:

❑ يشير المختبرون إلى مشاكل في البرامج

❑ يحب المطورون الاعتقاد بأن برامجهم مثالية

❑ يُنظر إلى المختبرين على أنهم يؤخرون المشروع عن طريق العثور على أخطاء في النظام

❑ يتوجب على المختبر إيصال النتائج التي وجدها بشكل موضوعي وليس بشكل شخصي

- It is important that they work together
- It is also important that they have mutual respect (احترام وتقدير متبادل) for each other.
- Collaboration (التعاون) is the right approach (النهج الصحيح) – **we work to a common goal!**



Testing Constraints

1. Limited Schedule & Budget (ميزانية)
2. Limited tester Skills



Testing Within the Software Lifecycle

1. Test Levels
2. Test Types
3. Test Technique
4. Regression Testing



