

Cơ Bản Về Kiểm Thử

Basic Of Testing – BOT



BATIZENS

Người trình bày:



Đoàn Thị Kim Nhung



BA Team

Mục Lục

01

Kiểm thử
Testing

02

Chất lượng phần mềm
Software Quality

03

Nguyên tắc kiểm thử
Software Testing Principles



BATIZENS



Từ ngữ viết tắt và định nghĩa

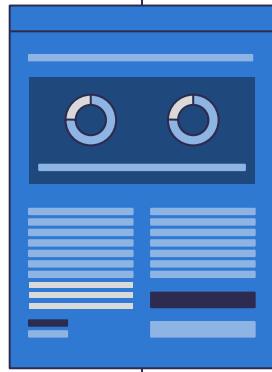
#	Từ ngữ viết tắt	Định nghĩa
1	PM	Phần mềm
2	CLPM	Chất lượng phần mềm
3		
4		
5		
6		
7		
8		





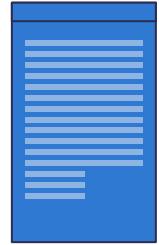
01

Kiểm thử





Kiểm thử



Khái niệm

Sao PM có lỗi?

Sao phải kiểm thử PM

Mục đích

Vai trò

Tầm quan trọng





1. Khái niệm

Kiểm thử phần mềm (Software Testing) là **hoạt động** nhằm **tìm kiếm** và **phát hiện** ra các **lỗi** của **phần mềm**, **đảm bảo** phần mềm **chính xác**, **đúng** và **đầy đủ** theo **yêu cầu** của **khách hàng**, yêu cầu của **sản phẩm** đã đặt ra.

Khái niệm

Sao PM có lỗi?

Sao phải kiểm
thử PM?

Mục đích

Vai trò

Tâm quan trọng





KT

2. Sao phần mềm có lỗi?

Khái niệm

Sao PM có lỗi?

Sao phải kiểm
thử PM?

Mục đích

Vai trò

Tâm quan trọng

Do con
người

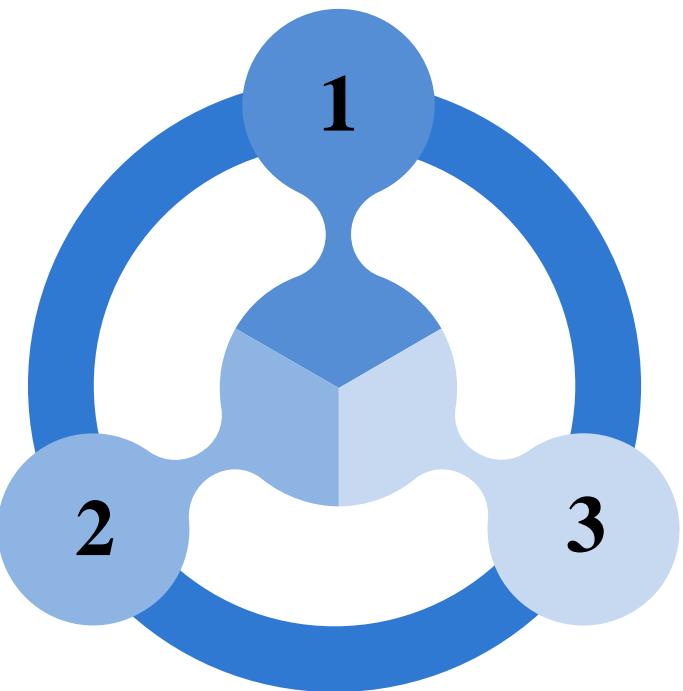
1

Do xác định nhu
cầu và phân tích

2

3

Do công cụ sử dụng
trong quá trình kiểm thử





KT

3. Sao phái kiểm thử phần mềm?

Kiểm thử phần mềm là một phần không thể thiếu trong quy trình phát triển phần mềm để đảm bảo rằng sản phẩm cuối cùng đáp ứng được yêu cầu và mong đợi của người dùng một cách đáng tin cậy và chất lượng.

Khái niệm

- Chỉ ra những khuyết điểm và sai sót đã được thực hiện trong giai đoạn phát triển.

• Đảm bảo độ tin cậy cho khách hàng và sự hài lòng của họ trong ứng dụng.

• Đảm bảo chất lượng của sản phẩm.

• Cần thiết cho hoạt động hiệu quả của ứng dụng phần mềm.

• Đảm bảo rằng các ứng dụng không có bất kỳ lỗi nào trong các giai đoạn trước-trong-sau của quá trình phát triển.

• Đó là yêu cầu thiết yếu giúp sản phẩm tồn tại trong kinh doanh.

Sao PM có lỗi?

Sao phái kiểm
thử PM?

Mục đích

Vai trò

Tâm quan trọng





KT

4. Mục đích

1. Tìm các lỗi được tạo ra bởi các lập trình viên khi phát triển phần mềm.
2. Đạt được sự tự tin và cung cấp thông tin về mức độ chất lượng.
3. Để ngăn ngừa lỗi.
4. Để đảm bảo rằng kết quả cuối cùng đáp ứng các yêu cầu kinh doanh và người sử dụng.
5. Để đạt được sự tín nhiệm của khách hàng bằng cách cung cấp cho họ 1 sản phẩm chất lượng.

Khái niệm

Sao PM có lỗi?

Sao phải kiểm
thử PM?**Mục đích**

Vai trò

Tâm quan trọng





KT

5. Vai trò

Khái niệm

Sao PM có lỗi?

Sao phải kiểm
thử PM?

Mục đích

Vai trò

Tâm quan trọng

1. Phát hiện lỗi và vấn đề

Một trong những vai trò quan trọng nhất của kiểm thử là phát hiện lỗi, sai sót và vấn đề trong phần mềm. Bằng cách thực hiện các ca kiểm thử khác nhau, những lỗi tiềm ẩn và không mong muốn có thể được phát hiện và báo cáo để sửa chữa.

2. Đảm bảo chất lượng

Kiểm thử phần mềm giúp đảm bảo rằng phần mềm được phát triển đạt được mức độ chất lượng mong đợi. Điều này bao gồm việc đảm bảo tính đúng đắn, đáng tin cậy và hiệu suất của phần mềm.

3. Xác nhận yêu cầu

Kiểm thử phần mềm giúp xác nhận rằng phần mềm đáp ứng đúng các yêu cầu đã được xác định từ các bên liên quan, bao gồm khách hàng và người dùng cuối.

Tối ưu hóa quá trình phát triển: Bằng cách phát hiện lỗi sớm và cung cấp phản hồi cho nhóm phát triển, kiểm thử phần mềm giúp tối ưu hóa quá trình phát triển bằng cách giảm thiểu thời gian và chi phí cần thiết để sửa chữa lỗi sau này.





KT

5. Vai trò

Khái niệm

Sao PM có lỗi?

Sao phải kiểm
thử PM?

Mục đích

Vai trò

Tâm quan trọng

4. Đảm bảo tương thích và tương thích

Kiểm thử phần mềm cũng đảm bảo rằng phần mềm hoạt động tương thích với các hệ thống, thiết bị và môi trường khác nhau mà nó có thể gặp phải trong quá trình triển khai và sử dụng.

5. Tăng cường tin cậy và uy tín

Việc thực hiện kiểm thử phần mềm một cách cẩn thận giúp tăng cường tin cậy và uy tín của sản phẩm trong mắt khách hàng và người dùng cuối.

6. Phản hồi và cải thiện

Cuối cùng, kiểm thử phần mềm cung cấp thông tin phản hồi quan trọng cho nhóm phát triển để cải thiện sản phẩm. Bằng cách xác định và báo cáo các lỗi và vấn đề, kiểm thử giúp nhóm phát triển hiểu rõ hơn về các điểm yếu của sản phẩm và đề xuất các cải tiến để tăng cường chất lượng và hiệu suất của phần mềm.





6. Tầm quan trọng

Khái niệm

Sao PM có lỗi?

Sao phải kiểm
thử PM?

Mục đích

Vai trò

Tầm quan trọng

1. Đảm bảo chất lượng sản phẩm

Kiểm thử phần mềm đảm bảo rằng sản phẩm cuối cùng đáp ứng được các tiêu chuẩn chất lượng và yêu cầu của người dùng. Bằng cách phát hiện và sửa chữa các lỗi và vấn đề, kiểm thử giúp tăng cường tính ổn định, đáng tin cậy và hiệu suất của phần mềm.

2. Giảm thiểu rủi ro

Kiểm thử phần mềm giúp giảm thiểu rủi ro liên quan đến sự cố và lỗi trong quá trình sử dụng phần mềm. Bằng cách phát hiện các lỗi sớm và sửa chữa chúng trước khi phần mềm được phát hành, kiểm thử giúp giảm thiểu nguy cơ gặp phải các vấn đề không mong muốn và giảm thiểu tác động tiêu cực đến người dùng và doanh nghiệp.

3. Tiết kiệm thời gian và chi phí

Mặc dù việc thực hiện kiểm thử có thể tốn kém về thời gian và tài nguyên, nhưng nó thường tiết kiệm thời gian và chi phí lớn hơn bằng cách phát hiện và sửa chữa các lỗi sớm. Việc sửa chữa lỗi sau khi phần mềm đã được triển khai có thể tốn kém và phức tạp hơn nhiều so với việc sửa chữa chúng trong quá trình phát triển.



KT

6. Tầm quan trọng

Khái niệm

Sao PM có lỗi?

Sao phải kiểm
thử PM?

Mục đích

Vai trò

Tầm quan trọng

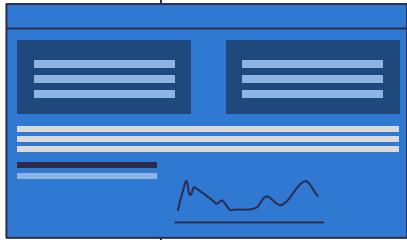
4. Tăng cường sự tin cậy và uy tín

Sự tin cậy và uy tín của phần mềm là yếu tố quan trọng trong việc thu hút và giữ chân khách hàng. Kiểm thử phần mềm giúp đảm bảo rằng sản phẩm hoạt động một cách đáng tin cậy và ổn định, từ đó tạo ra lòng tin từ phía người dùng và tăng cường uy tín của doanh nghiệp.

5. Cải thiện trải nghiệm người dùng

Kiểm thử phần mềm giúp đảm bảo rằng sản phẩm đáp ứng được nhu cầu và mong đợi của người dùng. Bằng cách kiểm tra và xác minh tính đúng đắn và hiệu suất của phần mềm, kiểm thử giúp cải thiện trải nghiệm người dùng và tạo ra một môi trường sử dụng sản phẩm thoải mái và thuận lợi.





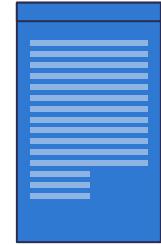
02



Chất lượng phần mềm



Chất lượng phần mềm



Khái niệm & các khía cạnh

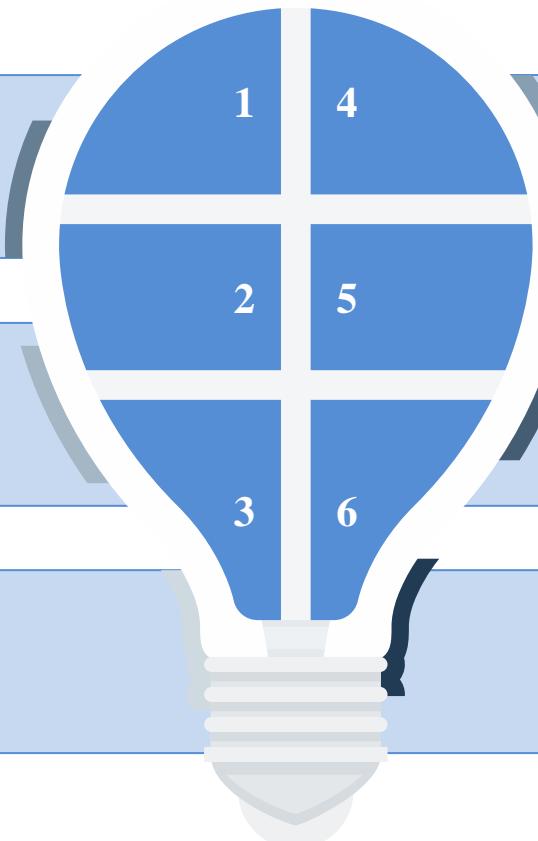
Khái niệm đảm bảo CLPM

Kiểm soát CLPM

So sánh SQA & SQC

Xác minh & Xác thực

Kiểm thử & Gỡ rối





1. Khái niệm và các khía cạnh

Khái niệm & các khía cạnh

Khái niệm đảm bảo CLPM

Kiểm soát CLPM

So sánh SQA & SQC

Xác minh & Xác thực

Kiểm thử & Gỡ rối

Khái niệm:

Chất lượng phần mềm (Software Quality) đề cập đến mức độ mà một sản phẩm phần mềm đáp ứng được yêu cầu và mong đợi của khách hàng và người sử dụng cuối cùng. Nó không chỉ đơn thuần liên quan đến việc sản phẩm hoạt động như thế nào, mà còn bao gồm các thuộc tính khác như tính đáng tin cậy, hiệu suất, dễ sử dụng, và khả năng mở rộng.

Chất lượng phần mềm là một yếu tố quyết định trong việc thành công của một dự án phần mềm và sự hài lòng của người dùng cuối cùng. Đảm bảo chất lượng phần mềm là một quá trình liên tục và đòi hỏi sự chú ý và nỗ lực từ phía nhóm phát triển.





1. Khái niệm và các khía cạnh

Khái niệm & các khía cạnh

Khái niệm đảm bảo CLPM

Kiểm soát CLPM

So sánh SQA & SQC

Xác minh & Xác thực

Kiểm thử & Gỡ rối

Các khía cạnh:

- Tính đúng đắn (Correctness):** Đây là khía cạnh quan trọng nhất của chất lượng phần mềm, đảm bảo rằng phần mềm thực hiện chính xác các chức năng và nhiệm vụ mà nó được thiết kế để thực hiện.
- Tính đáng tin cậy (Reliability):** Tính đáng tin cậy liên quan đến việc phần mềm hoạt động ổn định và không gây ra các lỗi không mong muốn trong quá trình sử dụng.
- Tính hiệu suất (Performance):** Khía cạnh này đánh giá hiệu suất của phần mềm trong việc xử lý dữ liệu và phản hồi với người dùng trong mọi điều kiện và tải công việc.
- Tính bảo mật (Security):** Tính bảo mật đảm bảo rằng phần mềm bảo vệ thông tin và dữ liệu của người dùng khỏi các mối đe dọa và tấn công từ bên ngoài.
- Tính dễ sử dụng (Usability):** Tính dễ sử dụng liên quan đến việc phần mềm có thân thiện với người dùng cuối, dễ sử dụng và dễ học.





1. Khái niệm và các khía cạnh

Các khía cạnh:

- 6. Tính mở rộng (Scalability):** Tính mở rộng đảm bảo rằng phần mềm có khả năng mở rộng để đáp ứng được nhu cầu tăng trưởng của doanh nghiệp hoặc người dùng.
- 7. Tính tương thích (Compatibility):** Tính tương thích đảm bảo rằng phần mềm hoạt động tốt trên nhiều nền tảng, thiết bị và môi trường khác nhau mà người dùng có thể sử dụng.
- 8. Tính tiên tiến (Maintainability):** Tính tiên tiến liên quan đến việc phần mềm dễ bảo trì và nâng cấp, với mã nguồn dễ đọc và hiểu, và tài liệu bảo trì đầy đủ.
- 9. Tính tích hợp (Interoperability):** Tính tích hợp đảm bảo rằng phần mềm có khả năng làm việc một cách hiệu quả với các hệ thống, ứng dụng hoặc dịch vụ khác.

Mỗi khía cạnh của chất lượng phần mềm đều đóng vai trò quan trọng trong việc đảm bảo rằng sản phẩm phần mềm đáp ứng được yêu cầu và mong đợi của người dùng cuối cùng, và mỗi khía cạnh này cần được xem xét trong quá trình phát triển và kiểm thử sản phẩm.



2. Khái niệm Đảm bảo CLPM (SQA)

Khái niệm & các khía cạnh

Khái niệm Đảm bảo CLPM

Khái niệm Kiểm soát CLPM

So sánh SQA & SQC

Xác minh & Xác thực

Kiểm thử & Gỡ rối

Đảm bảo chất lượng phần mềm (Software Quality Assurance - SQA) là quy trình và hoạt động được thực hiện trong quá trình phát triển phần mềm nhằm đảm bảo rằng sản phẩm phần mềm được sản xuất đạt được các tiêu chuẩn chất lượng mong đợi. Mục tiêu của SQA là đảm bảo rằng các quy trình, phương pháp, và tiêu chuẩn được tuân thủ để đảm bảo chất lượng của sản phẩm phần mềm.





3. Khái niệm Kiểm soát CLPM (SQC)

Khái niệm & các khía cạnh

Khái niệm đảm bảo CLPM

Kiểm soát CLPM

So sánh SQA & SQC

Xác minh & Xác thực

Kiểm thử & Gỡ rối

Kiểm soát chất lượng phần mềm (Software Quality Control - SQC) là quá trình kiểm tra và đảm bảo rằng sản phẩm phần mềm đáp ứng các tiêu chuẩn chất lượng đã được xác định trước đó. Khác với SQA (Đảm bảo chất lượng phần mềm), SQC tập trung vào việc kiểm tra sản phẩm phần mềm cụ thể để đảm bảo rằng nó đáp ứng các yêu cầu và tiêu chuẩn chất lượng.





4. So sánh SQA & SQC

Khái niệm &
các khía cạnh

Khái niệm đảm
bảo CLPM

Kiểm soát
CLPM

So sánh
SQA & SQC

Xác minh &
Xác thực

Kiểm thử &
Gỡ rối

Nội dung	SQA	SQC
So sánh các đặc điểm chung		
Định nghĩa	Quy trình và hoạt động được thực hiện trước và trong quá trình phát triển phần mềm để đảm bảo rằng sản phẩm đáp ứng các tiêu chuẩn chất lượng mong đợi.	Quá trình và hoạt động được thực hiện sau khi sản phẩm phần mềm đã được tạo ra để kiểm tra và đảm bảo rằng nó đáp ứng các tiêu chuẩn chất lượng đã xác định.
Đối tượng	Tập trung vào các quy trình, phương pháp, và tiêu chuẩn được sử dụng trong quá trình phát triển phần mềm.	Tập trung vào kiểm tra và đánh giá sản phẩm phần mềm cụ thể để đảm bảo rằng nó đáp ứng các yêu cầu và tiêu chuẩn chất lượng.
Thời điểm thực hiện	Thực hiện trước và trong quá trình phát triển phần mềm.	Thực hiện sau khi sản phẩm phần mềm đã được tạo ra.
Mục tiêu	Đảm bảo rằng các quy trình phát triển phần mềm tuân thủ và đáp ứng các tiêu chuẩn chất lượng.	Kiểm tra và đảm bảo rằng sản phẩm phần mềm cụ thể đáp ứng các yêu cầu và tiêu chuẩn chất lượng đã xác định.





4. So sánh SQA & SQC

Khái niệm &
các khía cạnh

Khái niệm đảm
bảo CLPM

Kiểm soát
CLPM

So sánh
SQA & SQC

Xác minh &
Xác thực

Kiểm thử &
Gỡ rối

Nội dung	SQA	SQC
Phạm vi	Toàn bộ quá trình phát triển phần mềm.	Sản phẩm phần mềm cụ thể.
Các hoạt động chính	<ul style="list-style-type: none"> - Lập kế hoạch và quản lý chất lượng. - Kiểm soát cấu hình. - Kiểm thử và đánh giá chất lượng. - Đánh giá và xác minh tiêu chuẩn. - Phản hồi và cải thiện. 	<ul style="list-style-type: none"> - Kiểm thử sản phẩm. - Kiểm tra mã nguồn. - Kiểm tra tư duy logic. - Kiểm tra cấu hình. - Đánh giá lỗi và sửa chữa.
Quan hệ với nhóm phát triển phần mềm	Tương tác chặt chẽ với nhóm phát triển phần mềm để đảm bảo tính hợp tác và tuân thủ các tiêu chuẩn chất lượng.	Thường độc lập với nhóm phát triển phần mềm để đảm bảo tính khách quan và độc lập trong việc kiểm tra và đánh giá chất lượng sản phẩm.
So sánh các khía cạnh		
Lập kế hoạch và quản lý kiểm thử	Đảm bảo rằng quy trình và kế hoạch kiểm thử được thiết lập và quản lý để đảm bảo rằng tất cả các yêu cầu và tiêu chuẩn chất lượng được đáp ứng.	Thực hiện lập kế hoạch và quản lý kiểm thử để đảm bảo rằng các hoạt động kiểm thử được thực hiện theo kế hoạch và đáp ứng các tiêu chuẩn chất lượng đã xác định.





4. So sánh SQA & SQC

Khái niệm &
các khía cạnh

Khái niệm đảm
bảo CLPM

Kiểm soát
CLPM

So sánh
SQA & SQC

Xác minh &
Xác thực

Kiểm thử &
Gỡ rối

Nội dung	SQA	SQC
Kiểm tra mã nguồn	Có thể kiểm tra mã nguồn nhưng không phải là hoạt động chính.	Thực hiện kiểm tra mã nguồn để tìm ra các lỗi cú pháp, lỗi logic và tiềm ẩn trong mã nguồn của phần mềm.
Kiểm tra tư duy logic	Thường không thực hiện kiểm tra tư duy logic trực tiếp, nhưng có thể tham gia vào quy trình đánh giá và đảm bảo logic của các quy trình phát triển.	Thực hiện kiểm tra tư duy logic để đảm bảo rằng logic của phần mềm là chính xác và phù hợp với yêu cầu.
Kiểm tra cấu hình	Có thể tham gia vào quy trình kiểm tra cấu hình nhưng thường không thực hiện trực tiếp.	Thực hiện kiểm tra cấu hình để đảm bảo rằng các cấu hình phần mềm được thiết lập và duy trì đúng cách.
Đánh giá lỗi và sửa chữa	Có thể tham gia vào quy trình đánh giá lỗi và phản hồi nhưng thường không thực hiện trực tiếp.	Thực hiện đánh giá lỗi và sửa chữa các lỗi phát sinh trong quá trình kiểm thử để cải thiện chất lượng của sản phẩm phần mềm.
	Có thể thấy rằng SQA thường tập trung vào việc đảm bảo rằng quy trình phát triển phần mềm đáp ứng các tiêu chuẩn chất lượng, trong khi SQC tập trung vào việc kiểm tra và đánh giá sản phẩm phần mềm cụ thể để đảm bảo rằng nó đáp ứng các yêu cầu và tiêu chuẩn chất lượng đã xác định.	





5. Xác minh & xác thực

Khái niệm & các khía cạnh

Khái niệm đảm bảo CLPM

Kiểm soát CLPM

So sánh SQA & SQC

Xác minh & Xác thực

Kiểm thử & Gỡ rối

Xác minh (Verification): Xác minh là quá trình kiểm tra xem sản phẩm phần mềm có đang tuân thủ các yêu cầu được ghi lại trong tài liệu không. Nó liên quan đến việc đảm bảo rằng phần mềm được xây dựng đúng cách, theo đúng các yêu cầu và thiết kế đã được đặt ra.

Trong quá trình xác minh, chúng ta thường hỏi: "Chúng ta đang làm đúng không?" và "Chúng ta đang xây dựng sản phẩm đúng cách không?".

Xác thực (Validation): Xác thực là quá trình kiểm tra xem sản phẩm phần mềm có đáp ứng được nhu cầu và mong đợi của khách hàng không. Nó liên quan đến việc đảm bảo rằng phần mềm đang được xây dựng là sản phẩm mà khách hàng thực sự cần và muốn.

Trong quá trình xác thực, chúng ta thường hỏi: "Chúng ta đang làm sản phẩm đúng không?" và "Sản phẩm có đáp ứng nhu cầu và mong đợi của khách hàng không?".

Xác minh là việc kiểm tra xem chúng ta đang xây dựng sản phẩm đúng cách không, trong khi xác thực là việc kiểm tra xem chúng ta đang xây dựng sản phẩm đúng không. Đây là hai khía cạnh quan trọng và bổ sung nhau trong quá trình đảm bảo chất lượng phần mềm.





6. Kiểm thử & gỡ rối

Khái niệm &
các khía cạnh

Khái niệm đảm
bảo CLPM

Kiểm soát
CLPM

So sánh
SQA & SQC

Xác minh &
Xác thực

Kiểm thử &
Gỡ rối

Kiểm thử (Testing):

- Kiểm thử là quá trình kiểm tra và đánh giá các tính năng, chức năng và hiệu suất của phần mềm để đảm bảo rằng nó hoạt động đúng và đáp ứng các yêu cầu đã đặt ra.
- Trong quá trình kiểm thử, các ca kiểm thử được thiết kế và thực thi để phát hiện ra lỗi và sự cố trong phần mềm, từ đó giúp cải thiện chất lượng của sản phẩm.
- Mục tiêu của kiểm thử là tìm ra các lỗi và đảm bảo rằng phần mềm đáp ứng được các tiêu chuẩn chất lượng.





6. Kiểm thử & gỡ lỗi

Khái niệm &
các khía cạnh

Khái niệm đảm
bảo CLPM

Kiểm soát
CLPM

So sánh
SQA & SQC

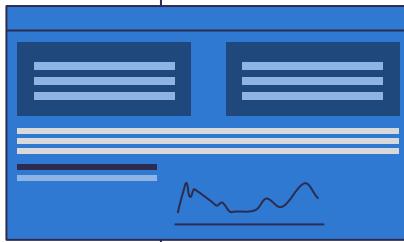
Xác minh &
Xác thực

Kiểm thử &
Gỡ lỗi



Gỡ lỗi (Debugging):

- Gỡ lỗi là quá trình xác định, phân tích và sửa chữa các lỗi và sự cố trong mã nguồn của phần mềm sau khi chúng đã được phát hiện bởi quá trình kiểm thử hoặc sau khi người dùng báo cáo về chúng.
 - Trong quá trình gỡ lỗi, lập trình viên sẽ xác định nguyên nhân của lỗi, điều này có thể là lỗi cú pháp, lỗi logic hoặc lỗi trong quá trình thực thi.
 - Mục tiêu của gỡ lỗi là tìm ra và sửa chữa các lỗi trong mã nguồn để sản phẩm có thể hoạt động một cách chính xác và đáp ứng nhu cầu của người dùng.
- Kiểm thử là quá trình kiểm tra và đánh giá chất lượng của phần mềm, trong khi gỡ lỗi là quá trình xác định và sửa chữa các lỗi đã được phát hiện trong quá trình kiểm thử hoặc trong quá trình sử dụng sản phẩm.



03



Nguyên tắc kiểm thử



Nguyên tắc kiểm thử

#	Tên nguyên tắc	Mô tả
1	Kiểm thử chứng minh sự hiện diện của lỗi (Testing shows the presence of defects)	<p>Ý nghĩa: Kiểm thử không thể chứng minh rằng phần mềm không có lỗi. Thay vào đó, nó chỉ có thể phát hiện và báo cáo về sự tồn tại của các lỗi.</p> <p>Giải thích: Dù kiểm thử có thể phát hiện một số lỗi, nhưng không thể chứng minh rằng không có lỗi nào tồn tại trong phần mềm. Có thể có nhiều lỗi còn chưa được phát hiện hoặc một số lỗi không thể kiểm thử được.</p>
2	Kiểm thử toàn bộ là không khả thi (Exhaustive testing is not possible)	<p>Ý nghĩa: Không thể kiểm tra tất cả các trường hợp và điều kiện có thể xảy ra trong một phần mềm.</p> <p>Giải thích: Với số lượng trường hợp sử dụng và điều kiện đầu vào có thể vô hạn, việc kiểm thử tất cả các trường hợp là không khả thi. Thay vào đó, kiểm thử tập trung vào các trường hợp quan trọng nhất và có thể gây ra nhiều vấn đề nhất.</p>





Nguyên tắc kiểm thử

#	Tên nguyên tắc	Mô tả
3	Kiểm thử càng sớm càng tốt (Early testing)	<p>Ý nghĩa: Kiểm thử nên được tiến hành từ giai đoạn phát triển sớm nhất của quy trình phát triển phần mềm.</p> <p>Giải thích: Phát hiện lỗi sớm giúp giảm thiểu chi phí sửa chữa và tối ưu hóa quy trình phát triển bằng cách phát hiện và khắc phục các vấn đề ngay từ đầu.</p>
4	Lỗi phân bố tập trung (Defect clustering)	<p>Ý nghĩa: Trong một phần mềm, số lượng lỗi thường tập trung ở một số ít các khu vực quan trọng nhất.</p> <p>Giải thích: Có một số ít chức năng hoặc khu vực mã nguồn có thể chứa nhiều lỗi hơn so với các phần khác. Tập trung kiểm thử vào những khu vực này có thể mang lại hiệu suất cao hơn trong quá trình kiểm thử.</p>





Nguyên tắc kiểm thử

#	Tên nguyên tắc	Mô tả
5	Nghịch lý thuốc trừ sâu (pesticide paradox)	<p>Ý nghĩa: Nếu sử dụng cùng một phương pháp kiểm thử lâu dài, nó sẽ không còn phát hiện được nhiều lỗi mới.</p> <p>Giải thích: Nếu kiểm thử không được cập nhật và cải tiến, nó có thể trở nên không hiệu quả và bỏ qua các lỗi mới hoặc tiềm ẩn. Do đó, phương pháp kiểm thử cần được đánh giá và cập nhật định kỳ.</p>
6	Kiểm thử phần mềm phụ thuộc vào ngữ cảnh (Testing is Context-Dependent)	<p>Ý nghĩa: Phương pháp và phạm vi kiểm thử phụ thuộc vào ngữ cảnh, yêu cầu và môi trường cụ thể của dự án phần mềm.</p> <p>Giải thích: Không có một phương pháp kiểm thử đúng cho tất cả các dự án. Kiểm thử cần phải điều chỉnh và tinh chỉnh dựa trên ngữ cảnh cụ thể của từng dự án để đạt được hiệu suất cao nhất.</p>





Nguyên tắc kiểm thử

#	Tên nguyên tắc	Mô tả
7	Quan niệm sai lầm về việc “hết lỗi” (Absence of Errors fallacy)	<p>Ý nghĩa: Mặc dù kiểm thử có thể tìm ra nhiều lỗi, nhưng không thể đảm bảo rằng phần mềm không có lỗi.</p> <p>Giải thích: Kiểm thử chỉ có thể chứng minh rằng có lỗi xuất hiện, nhưng không thể chứng minh rằng không có lỗi. Mặc dù kiểm thử có thể cải thiện chất lượng của phần mềm, nhưng không bao giờ có thể đảm bảo rằng nó là hoàn hảo.</p>





B A T I Z E N S



Trân trọng cảm ơn!

