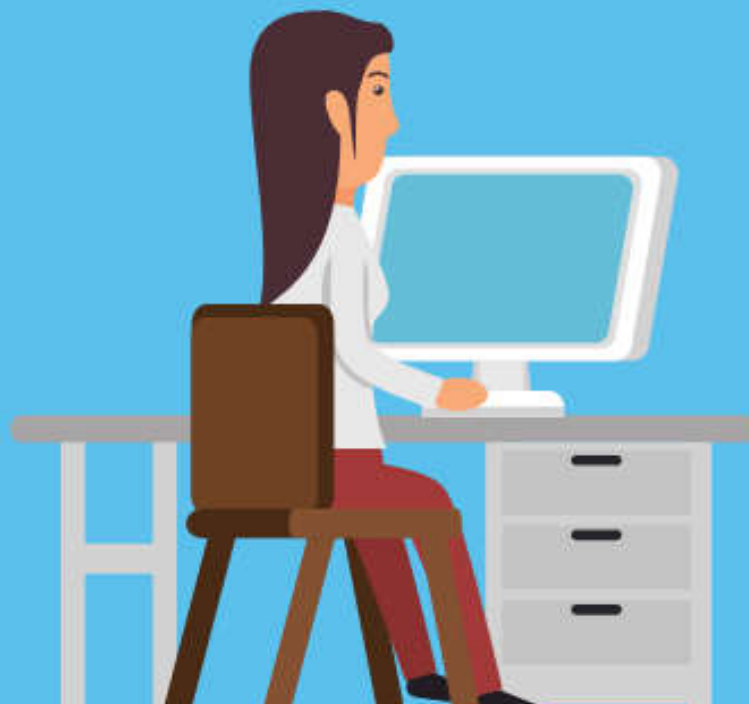


TEST IO & UTEST PLATFORMS

# TESTING ONLINE WORK

BY DANG PHUONG  
TESTERTUDO TEAM

test IO    UTest®



# Kiểm Tiền Testing Online- Kiểm thử website và mobile app dành cho Newbies

*Testertudo Team*

*Đặng Phương*



Copyright © 2024 by Dang Phuong.

All rights reserved.

# Giới Thiệu

Xin chào mọi người!

Mình là Đặng Phương. Hiện mình đang làm việc online như là một Freelance tester với hơn 6 năm kinh nghiệm trên các nền tảng testing online như uTest, Test IO. Bên cạnh đó như mọi người cũng biết mình hiện cũng đang là Admin của group Facebook Cộng đồng Freelance Tester Vietnam và là Leader của Testertudo team hoạt động trong mảng testing online.

Mình tiếp cận với công việc testing online ban đầu như một cách để tăng thêm nguồn thu nhập trong khoảng thời gian rảnh rỗi và mình nhận thấy được tiềm năng của công việc này là rất lớn đối với các bạn muốn tiến sâu trong lĩnh vực MMO và kiếm tiền online. Nguồn thu nhập có thể là hàng trăm \$ hay vài ngàn \$ hoàn toàn trong tầm tay nếu bạn thực sự nghiêm túc với công việc. Mình đã theo đuổi công việc testing online trong nhiều năm qua và đã tích lũy được khá nhiều kinh nghiệm khi làm việc trong lĩnh vực này. Đó sẽ là những điều mà mình muốn chia sẻ trong cuốn Ebook này.

## • Bạn Sẽ Học Được Gì Từ Bộ eBook Này.

- 👉 Hiểu về quá trình kiếm tiền từ công việc testing online.
- 👉 Biết về cách làm testing online đối với những người chưa có kinh nghiệm.
- 👉 Biết cách kiếm tiền online trên các nền tảng test Test IO và uTest
- 👉 Biết thêm về các tool, app và các tips hữu ích cho công việc testing online.

- **Testing online là gì**

Có rất nhiều các nền tảng test online để kiếm tiền dành cho tester nhưng Test IO và uTest là 2 website nổi bật nhất mà trong suốt nhiều năm qua các tester online trên thế giới đang kiếm tiền online từ công việc này. Vậy testing online là gì?




Để hiểu một cách đơn giản thì đây là công việc kiểm thử thủ công website, ứng dụng mobile. hay nói cách khác Tester có nhiệm vụ là sử dụng các website, mobile app ( được Test IO hoặc uTest cung cấp) và tìm ra các lỗi (bug) bất thường trên đó. Việc tìm ra được lỗi có giá trị thì tester sẽ nhận được tiền cho mỗi bug đó. Giá trị của bug thường sẽ là từ €2- €15/bug hoặc có những dự án giá trị bug có thể lên tới >€20

Ngoài việc tìm bug thì tester có thể kiếm được thêm tiền dựa vào các công việc như tái hiện lỗi ( reproduce bug), làm test case, user story... nhưng nguồn chính vẫn là test tìm bug.

Theo mình thấy thì số tiền mà một tester online ở mức khá làm thêm trong khoảng thời gian free-time 2-4h/ngày thì con số có thể đạt 200-600Euro/tháng hoặc hơn nữa tùy thuộc vào kỹ năng của tester và số lượng dự án của tháng đó. Tuy con số không quá cao nhưng theo mình đánh giá thì so với mặt bằng thu nhập ở VN thì một công việc làm trong khoảng thời gian free-time, không bỏ chi phí đầu tư và đặc biệt là có thể làm ở bất cứ đâu chỉ với một chiếc laptop và smartphone kết nối mạng thì đây là một công việc khá thú vị phải không ạ.

Thường thì một dự án test ( test cycle) sẽ diễn ra trong khoảng từ 2-4 ngày do vậy tester có thể lựa chọn thời gian phù hợp để tham gia test ( Dĩ nhiên trong dự án sẽ có các tester khác tham gia cùng nên việc làm sớm nhất sẽ càng có lợi trong việc tìm thấy bug và submit bug sớm nhất)

### Payout

Bug Type	Amount	Reprod.
 Low	€15.00	€0.50
 High	€17.00	€0.50
 Critical	€20.00	€0.50

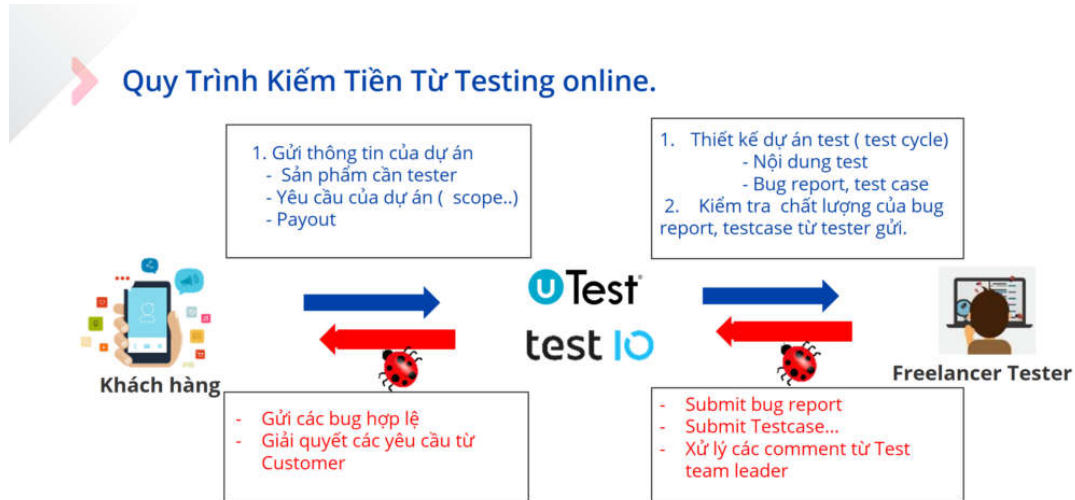
#### Bug Payouts

BUG TYPE	BASE PAYOUT	VERY VALUABLE	EXCEPTIONAL
Visual	\$3.00	\$18.00	\$30.00
Functional	\$3.00	\$18.00	\$30.00
Crash	\$3.00	\$18.00	\$30.00
Content	\$1.50	\$16.50	\$28.50
Performance	\$3.00	\$18.00	\$30.00

#### Your Specified Environments

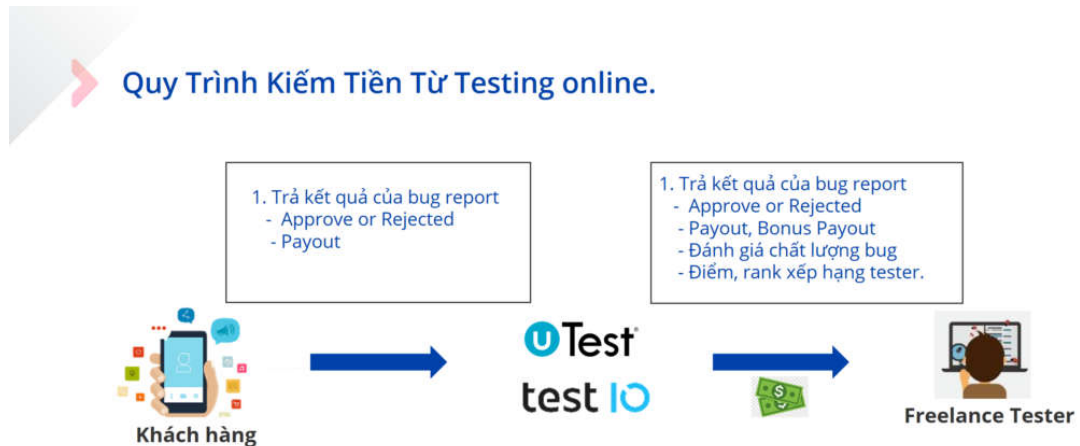
- **Hiểu rõ hơn về quy trình làm và nhận tiền của freelance tester từ mảng này như thế nào?**

Quy trình 1: Nhận và triển khai dự án



Quy trình 1: Bắt Đầu Dự Án

Quy trình 2: Trả kết quả+ kết thúc dự án



Quy trình 2: Kết thúc +Thanh toán

- **Tester nhận tiền testing online bằng cách nào.**

Do các platform test như uTest, Test IO đều là của nước ngoài nên tester sẽ nhận tiền từ công việc test thông qua các ví điện tử Payoneer hoặc Paypal.

Tiền test sẽ được trả hàng tháng về tài khoản Payoneer/Paypal mà tester đã liên kết với tài khoản sau đó tester có thể rút ngoại tệ trong các ví điện tử đó về ngân hàng nội địa

- **Đối tượng phù hợp tham gia mảng testing online này:**

Đây là một công việc mà ai có đam mê về công nghệ ( máy tính, smartphone) thì hoàn toàn có thể thử sức mình vì thực tế cộng đồng freelance tester có số lượng tester ngoài ngành chiếm khá lớn.

Hiểu đơn giản hơn thì với testing online, tester sẽ đóng vai trò như là những người dùng thông thường để sử dụng, trải nghiệm các sản phẩm của khách hàng mà Test IO hoặc uTest cung cấp và tìm ra các lỗi có giá trị để được nhận thanh toán. Các sản phẩm mà tester test sẽ đa phần giống với website hay ứng dụng thường ngày mà chúng ta đang sử dụng như: Shopee, Tiki, các ứng dụng, game điện thoại trên IOS/ android.. và việc chúng ta thường gặp các lỗi như hình ảnh hay chữ bị broken, cắt góc hay không thực hiện được một vài thao tác trên web, app hoặc khi đang dùng app thì tự nhiên bị văng ra màn hình chính.v.v.v rất nhiều trường hợp nữa mà chúng ta gặp hằng ngày nhưng không để ý đó chính là các lỗi mà khách hàng họ đang rất cần biết để fix giúp hoàn thiện chất lượng sản phẩm của họ hơn và nhiệm vụ của freelance tester như chúng ta là thông tin cho họ biết cụ thể về lỗi đó thông qua một báo cáo lỗi hay còn gọi là " bug report "

Bản thân mình đánh giá công việc này đòi hỏi kiến thức không quá khó nhưng để làm tốt và thành công thì đòi hỏi tester phải có đam mê, nỗ lực cải thiện các tư duy tìm lỗi, kỹ năng tiếng anh để đi xa và hướng tới các vị trí cao trong các platform testing với mức thu nhập ngàn \$. ( Hiện nay ở VN đang có nhiều freelance tester đang nắm các vị trí cao trong các platform test như uTest và Test IO.

Kết lại theo mình thì để thử sức với mảng kiếm tiền testing online bạn cần tối thiểu đáp ứng với những tiêu chí sau:

1. Có đam mê và nhanh nhạy về công nghệ: Sử dụng smartphone, máy tính
2. Có thiết bị test: ít nhất là một chiếc máy tính và một smartphone hệ điều hành Android/ IOS ( không bắt buộc phải dùng đời cao nhưng ít nhất thiết bị vẫn còn hỗ trợ hệ điều hành mới nhất)
3. Có khả năng đọc viết tiếng anh ở mức cơ bản để hiểu yêu cầu của dự án và viết bug report ( Dĩ nhiên là quá trình làm chúng ta có thể sử dụng các công cụ hỗ trợ dịch để làm)
4. Cuối cùng là chăm chỉ nỗ lực là yếu tố không bao giờ được thiếu.

Để các bạn newbie không quá hoang mang thì theo khảo sát trong group cộng đồng freelance tester thì đa phần các tester online kiếm tiền từ mảng này lại không phải làm nghề tester tại công ty cho nên không có lý do gì mà chúng ta lại không thử sức mình để tạo thêm một nguồn thu nhập online trong khoảng thời gian free-time

- **Kiểm \$1000/tháng từ công việc này có được không?**

Câu trả lời là **Có** thậm chí có những tháng con số còn hơn nhưng phụ thuộc vào nhiều yếu tố: kỹ năng tester, lượng dự án tốt trong tháng đó được gửi về...

Bản thân mình đã nhiều năm trong mảng này với vai trò là freelance tester thì mình khuyên thật các bạn nên bắt đầu làm testing online với tâm thế ban đầu là nghề tay trái, dành 2-4h free-time/ngày để làm mảng này với mục đích kiếm thêm nguồn thu nhập đặc biệt là trong giai đoạn nguồn thu nhập chính thấp hoặc bị ảnh hưởng sau đó mới nghĩ đến hướng phát triển là trở thành các test team leader ( TTL) của các nền tảng test: Test IO, uTest để được nhận những mức thù lao cao với công việc làm online.

Testing online là một hình thức kiếm tiền đã xuất hiện từ lâu trên thế giới, hầu hết các công ty testing online lớn đến từ Mỹ, Châu Âu. Trên thực tế nguồn nhân lực Tester chưa đáp ứng đủ so với nhu cầu trên thị trường do vậy hình thức testing online này đã ra đời nhằm tận dụng tối đa nguồn lực của các Freelance

tester đến từ khắp nơi trên khắp nơi để tham gia các dự án kiểm thử các ứng dụng, game mobile hay website cho các khách hàng, và dĩ nhiên với mỗi dự án hoàn thành thì các tester sẽ được trả tiền từ công ty và số tiền này sẽ được trả về các ví điện tử Paypal hoặc Payoneer hàng tháng.

Số tiền mà các tester kiếm được là khác nhau và phụ thuộc vào khả năng bắt lỗi (bug) và chất lượng bug report nhưng nó rất là xứng đáng với thời gian mà tester bỏ ra khi tham gia làm việc trên các dự án.

- **Cần chuẩn bị gì để làm được công việc testing online.**

Mọi người thường nghĩ chỉ học về tester mới làm được testing online nhưng nhận định đó là sai lầm! Vì mình biết rất nhiều bạn không có background về IT vẫn làm được testing online thậm chí rất thành công. Vậy cần chuẩn bị những gì để làm testing online:

1. **Thiết bị test:** Bạn cần có ít nhất 2 loại thiết bị như sau để thực hiện công việc testing online: Smartphone (iOS/ Android) và một chiếc PC/Laptop.
2. **Tiếng anh:** Làm việc trên các nền tảng của các công ty nước ngoài do vậy tiếng anh là điều mà các bạn cần chuẩn bị. Không đòi hỏi kiến thức chuyên sâu nhưng tối thiểu bạn phải đọc hiểu được email, nội dung bài test bằng tiếng anh (Có thể sử dụng các công cụ hỗ trợ dịch trong quá trình làm việc).
3. **Có đam mê về thiết bị công nghệ:** Điều này giúp bạn giữ được lửa khi làm test.
4. **Có tài khoản ngân hàng:** Mục đích để bạn có thể nhận và rút được tiền trả từ khách hàng cho công việc testing online.

# Cách Làm Testing Online Với Những Người Không Có Kinh Nghiệm.

## Các kiến thức mà các bạn cần chuẩn bị để làm testing online là gì?

**Note:** *Mình sẽ chỉ tập trung vào các điểm trọng tâm mà các bạn cần chú ý để làm testing online, trên thực tế để các bạn trở thành một tester và làm cho công ty thì cần học nhiều thêm thứ nữa nhưng mình sẽ không đề cập tới vì nó khá rộng và không nằm trong phạm vi cuốn Ebook này.*

### 1. Các thuật ngữ phổ biến mà các bạn nhập môn cần biết:

*Dưới đây là những khái niệm mà mình lựa chọn giúp các bạn chưa có kiến thức về testing dễ hiểu nhất để phục vụ cho công việc testing online.*

- **Bug (Lỗi):** Là những trường hợp diễn ra không đúng theo thiết kế của sản phẩm gây cản trở người dùng sản phẩm đó.
- **Bug report (Báo cáo lỗi):** Mô tả lỗi xảy ra khi thực thi test sản phẩm giúp khách hàng nắm được quá trình xảy ra lỗi và các thông tin cần thiết về lỗi đó.
- **Test cycle ( chu kỳ test):** là một dự án test mà công ty gửi tới trong đó có chứa thông tin về sản phẩm cần test, khách hàng mà ở đó các tester sẽ thực hiện báo cáo bug tại cycle đấy và trao đổi với test leader hoặc các tester cùng tham gia cycle này.
- **Test case:** là 1 tập hợp các trường hợp điều kiện theo đó mà Tester có thể dựa vào nó để xác định liệu 1 ứng dụng, hệ thống phần mềm hoặc là 1 trong các tính năng của nó có hoạt động như mong muốn cần làm hay không?

## 2. Phân loại testing

Có nhiều loại testing mà các tester có thể chọn để tham gia testing online như sau: Funtional testting, Locazilation testing, Usability testing và Security testing. Trong đó theo mình thấy số lượng dự án Functional test cycle chiếm trên 90% tổng số lượng các dự án được gửi cho tester.

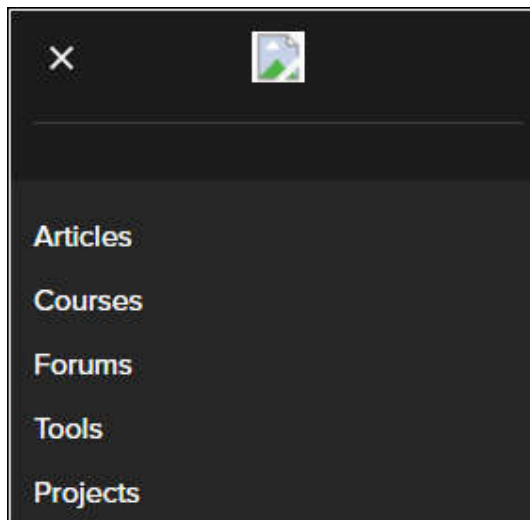
- **Functional testing là gì?** Hiểu một cách đơn giản kiểm thử chức năng là một loại kiểm thử nhằm xác minh rằng mỗi chức năng của sản phẩm được test được hoạt động đúng theo yêu cầu.

## 3. Phân loại Bug (Bug type)

### 3.1 Phân loại Bug Theo Nền Tảng uTest

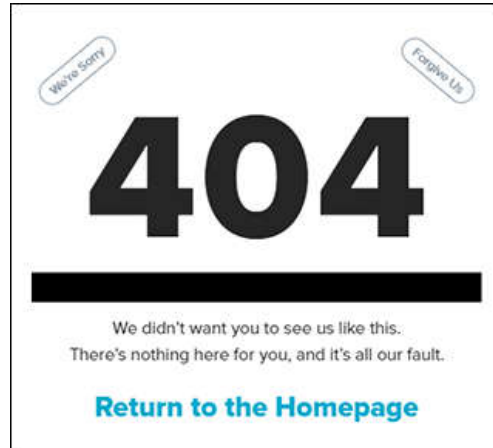
Theo uTest thì được chia ra làm 5 loại như sau:

- **Visual bug:** Là những lỗi liên quan đến đồ họa, giao diện sử dụng của sản phẩm. (Ví dụ: hình ảnh bị mất một phần hoặc chồng chéo lên nhau)



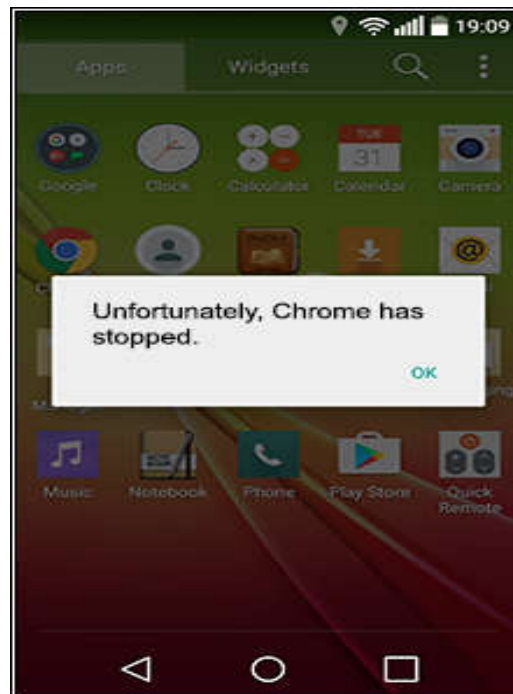
*Visual bug*

- **Functional bug**: Là lỗi mà ở đó hoạt động của ứng sản phẩm không theo như mong muốn của người sử dụng. (Ví dụ: Nút đăng nhập không hoạt động khi ấn vào)



*Functional bug*

- **Crash bug**: kịch bản lỗi làm cho ứng dụng bị sập hoặc bị bung ra ngoài mong muốn.



*Crash bug*

- **Content bug:** Là các lỗi liên quan đến ngữ pháp, chính tả hoặc chữ bản địa hóa. Ví dụ bug: The "Sep" word miss translated in Vietnamese

- **Performance bug:** Các lỗi tác động đến việc làm chậm hoạt động của ứng dụng và người sử dụng. (Ví dụ: App bị đơ, lác..)

### 3.1 Phân loại Bug Theo Nền Tảng Test IO

Hiện tại có 4 loại lỗi được công nhận trên nền tảng Test IO

- **Lỗi tính năng – Fuctional:** Đây là loại lỗi phổ biến nhất và 90% các cycle trên TestIO đều liên quan tới loại lỗi này. loại lỗi này liên quan tới tính năng của website/ phần mềm mà chúng ta được yêu cầu test. Lỗi tính năng có thể là việc bạn không đăng nhập vào được hệ thống, tính năng search trả về kết quả không mong muốn, ứng dụng bị crash. Nếu khi tương tác với website và ứng dụng mà kết quả trả về thì **có thể** đó là một lỗi tính năng.
- **Lỗi giao diện – Visual:** Lỗi liên quan tới giao diện của website và phần mềm.
  - Link lỗi hoặc image lỗi (404) trong các thành phần không liên quan tới tính năng của chúng (Nếu lỗi 404 ở phần navigaiton menu thì sẽ có la fuctional bug)
  - Thiếu nội dung, tooltip trống
  - Thiếu nội dung, phần hiển thị bị để trống
  - Dịch thiếu, khi site là tiếng Đức mà chữ lại là tiếng Anh
  - Nội dung trùng lặp
- **Đề xuất cải thiện – Usability suggestion:** Đây thực tế không phải là lỗi, nếu bạn có đề xuất để website/ứng dụng tốt hơn thì đấy chính là nó. Đây là một loại ticket đặc biệt, bạn sẽ không phải tìm lỗi của website/phần mềm mà là tìm những điểm cải tiến mà bạn cảm thấy cần thiết nên có. Dựa trên đề xuất của bạn, khách hàng sẽ quyết định để cải tiến các tính năng hiện có của

phần mềm để mang lại trải nghiệm tốt hơn cho người dùng. Điểm độc đáo của loại ticket này là

#### 4. Đánh giá mức độ nghiêm trọng của bug (Bug severity)

- **Low bug:** Là lỗi từ mấy function hiển thị và có phản hồi khi người dùng tương tác nhưng không quan trọng mà thích gây chú ý bằng việc nó có bug nhưng không ảnh hưởng đến mục đích chính của Web/App

Ví dụ: Review trong PDP không sort khi chọn condition, không filter theo Rating. Ballon của tooltip không hiện khi click vào.

- **High bug:** Là bug có ảnh hưởng đến mục đích chính của trang web, nhưng có thể thay thế bằng một cách khác để không bị bug, hoặc function có hiển thị trên web nhưng khi tương tác thì không có phản hồi.

Ví dụ: Bạn không xóa ở Cart được, nhưng vào Checkout bạn có thể xóa được. Không add ở PLP được nhưng vào PDP bạn có thể add được. Cẩn thận hén, dù sao cũng chỉ bị down từ crit xuống high thôi, cứ log crit đi

- **Critical Bug:** Là cái bug mà là chết Web/App, ảnh hưởng trực tiếp đến chức năng chính của Web/App nhưng không thể thay thế bằng con đường nào khác.

#### 5. Cách viết một báo cáo bug hoàn chỉnh (Bug report)

Vậy khi tester tìm ra lỗi trong quá trình sử dụng sản phẩm thì làm cách nào để chúng ta thông tin cho khách hàng biết được bug đó xảy ra như thế nào. Viết một báo cáo mô tả lỗi đó là cách để làm điều đó hay còn gọi là bug report.

Dưới đây là một form bug report chung mà các tester cần biết.

## Submit a Bug for \_\_\_\_\_

Feature	Landing Page	▼	
Bug Type	Functional	▼	
Severity	<input type="radio"/> Low	<input type="radio"/> High	<input type="radio"/> Critical
Title	<input type="text" value="Provide a short, descriptive title for your bug"/>		
URL	<input type="text" value="Provide the URL where your bug occurred"/>		
Steps	1 <input type="text"/>	↑ ↓ <input type="text"/>	🗑️
	+ Add a step		
Actual result	<input type="text" value="Describe the actual result or behaviour for this bug"/>		
Expected result	<input type="text" value="Describe the result or behaviour you expected for this bug"/>		
Upload attachment(s) <small>i</small>	<input type="text" value="+ Add attachment"/>		
Devices used	<input type="text" value="Apple iPhone SE (2020)"/>		

### Form bug report chung

#### Severity:

- Thể hiện mức độ khẩn cấp của 1 bug.
- Chỉ hiển thị ở Function

#### Title:

- Phải tóm tắt vấn đề, để người đọc có được ý tưởng chung về lỗi chỉ bằng cách đọc tiêu đề lỗi.
- Không cần phải đọc toàn bộ báo cáo để hiểu vấn đề là gì

- Chính xác và không quá dài.
- Tiêu đề bao gồm các thành phần: **Yếu tố liên quan, chức năng nào, nguồn gốc**

Ví dụ:

- Login button does not work -> sai
- The login page is not triggered after user clicks Login button. -> Đúng

#### **URL:**

- Đối với website thì đây là nơi xuất hiện lỗi, chỉ cần copy URL trên trình duyệt vào.
- Còn nếu là APP thì chỉ cần ghi là N/A hoặc để trống.

Ví dụ:

<https://website.com/login>

#### **Steps:**

- Đây là nơi dùng để mô tả tất các step để có thể dẫn tới việc tạo ra bug.
- Câu đầu tiên luôn là Open <https://website.com> nếu là web, còn app thì là Open {name} app

Ví dụ

Step 1: Open <https://website.com>

Step 2: Input your account into fields

Step 3: Click Login button or press Enter key.

#### **Actual:**

- Đây là một trong những trường quan trọng của report
- Bạn phải mô tả càng chi tiết càng tốt, đừng nói chung chung.
- Bạn thêm bất kì thông tin nào có liên quan đến lỗi: Bug chỉ xảy trên trên IE 11. Để đảm bảo người đọc hiểu quá trình bạn làm việc

**Ví dụ:** The user cannot redirect to other page after the user click Login button.

#### **Expected:**

- Mô tả một điều được cho là đúng dẫn được xảy ra sau khi thực hiện bước cuối cùng.
- Là một điều đáng lẽ phải xảy ra.

**Ví dụ:** The user should be redirected to the Homepage or receive an error message, for example, your account is locked, or your username or password is wrong after the user clicks the Login button.

**Note:** Actual và Expected không bao giờ được đối lập nhau. Hai nội dung này phải khác nhau rất nhiều

**Attachment:**

- Hầu hết các bug function điều yêu cầu 1 đoạn video
- Trừ một số trường hợp có thêm các đính kèm khác theo yêu cầu như: crash app thì cần thêm file log, bug visual và content thì bao gồm 1 hình ảnh, 1 video hoặc 1 hình thời gian và 1 hình bug

# Các Tool, App Hữu Ích Phục Vụ Công Việc Testing Online

Để phục cho công việc viết bug report thì các tester sẽ rất cần đến các app, tool hỗ trợ.

Dưới đây mình sẽ tổng hợp lại các app vô cùng hữu ích dành cho công việc testing online.

## 1. Chụp màn hình

Hầu hết các file ảnh yêu cầu đính kèm phải được định dạng về đuôi .jpg và đánh dấu vị trí xảy ra lỗi bằng cách vẽ 1 ô chữ nhật đỏ bao quanh vị trí lỗi.

- **Máy tính:** Chúng ta có thể đơn giản sử dụng phím tắt printer screen (PrtSc) kết hợp với ứng dụng Paint có sẵn để chỉnh lại.
- **Smart phone:** Nhấn tổ hợp phím chụp màn hình điện thoại là một trong những cách đơn giản để thực hiện

## 2. Quay màn hình

Các video đính kèm yêu cầu xuất ra file .mp4, dung lượng không quá cao (thường <25MB) và xóa âm thanh không cần thiết.

Note: Khi quay màn hình trên thiết bị android và Desktop thì cần bật con trỏ chuột hoặc show tap. Còn đối với iOS thì không bắt buộc phụ thuộc vào yêu cầu của từng test cycle.

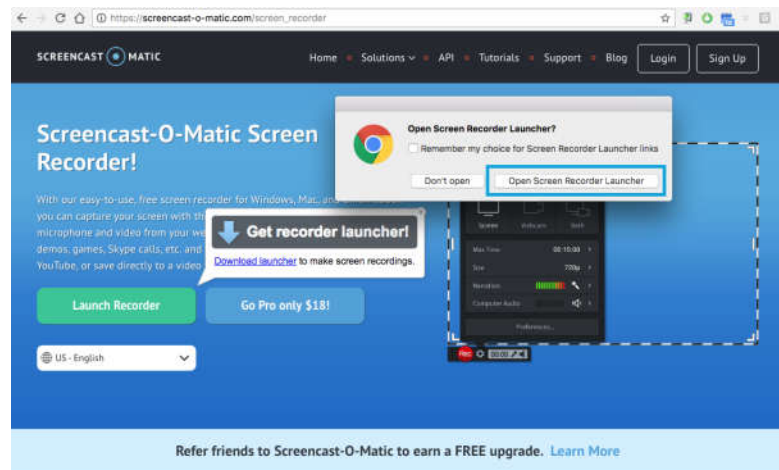
- **Máy tính:** Sử dụng ứng dụng Screencast-O-Matic, Bandicam

Mọi người có thể vào đây để download free: <https://screencast-o-matic.com/screen-recorder>

Hoặc có thể sử dụng tới tool Screen presso bản crack:

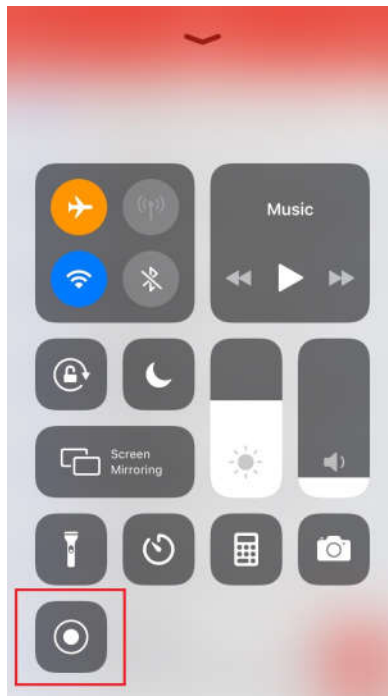
Link download:

<https://drive.google.com/file/d/1qFkYo2RfHjFyqnFZwodaeJKyFzwtZiPi/view>



### Phần mềm Screen-O-Matic

- **iPhone:** Đối với hệ điều hành iOS 11.x thì đã tích hợp sẵn quay màn hình video.



### Ứng dụng quay màn hình trên ios 11+

+ Hoặc có một cách khác nữa đó là chiếu màn hình điện thoại lên máy tính rồi ghi lại để xuất ra video. Chúng ta có thể sử dụng một phần mềm cài đặt trên máy tính đó là Lonely screen



*App Lonely screen*

Link download: <https://as.lonelyscreen.com/download.html>

### - Smart phone Android

Phần mềm AZ screen recorder ( ver: 2.92) là một trong những lựa chọn tối ưu cho việc quay màn hình. ( Hiện tải bản update trên CH Play đã bị cắt bỏ mục show tap trên video). Link download file cài đặt APK trên máy android: [https://drive.google.com/file/d/1wgqje5u9GdKtzO9TAwj8\\_zN965VTUI51/view?us](https://drive.google.com/file/d/1wgqje5u9GdKtzO9TAwj8_zN965VTUI51/view?us)



*App AZ Screen recorder*

- Vậy một câu hỏi đặt ra đó là sau khi quay màn hình mà video có dung lượng lớn và nhiều tiếng ồn thì làm cách nào để giải quyết điều này. Ứng dụng trên PC: **Handbrake** sẽ giải quyết điều này.

Link download: <https://handbrake.fr/>



*Phần mềm Handbrake*

### 3. Lấy file log

Với các lỗi crash thì khách hàng thường yêu cầu bắt buộc các tester đính kèm file log trên thiết bị khi gây ra lỗi đấy. Ngoài ra các lỗi functional có mức độ nghiêm trọng cao cũng thường yêu cầu đính kèm file log này.

```
deniedPermissions.permissions.photosPermission = 'denied';
deniedPermissions.preferences.save_photos_locally = false;

const deniedWrapper = shallow(
  <CameraSettings />,
  { context: { store: mockStore(deniedPermissions) } },
);
const deniedRender = deniedWrapper.dive();
deniedRender.find('Switch').forEach(child => {
  child.simulate('valueChange', true);
});

expect(saveToPhoneSpy.calledTwice).toBe(true);

// change photo permissions to undetermined
const undeterminedPermissions = { ...initialState };
undeterminedPermissions.permissions.photosPermission = 'undetermined';
undeterminedPermissions.preferences.save_photos_locally = false;

const undeterminedWrapper = shallow(
  <CameraSettings />,
  { context: { store: mockStore(undeterminedPermissions) } },
);
const undeterminedRender = undeterminedWrapper.dive();
undeterminedRender.find('Switch').forEach(child => {
  child.simulate('valueChange', true);
});
```

*Lấy file console log*

File log sẽ được lưu thành đuôi .txt

- **Lấy console log trên trình duyệt Chrome:**

Link video hướng dẫn: <http://bit.ly/2XPoJ00>

- **Lấy console log trên trình duyệt Firefox:**

Link video hướng dẫn: <http://bit.ly/2xCcm8M>

- **Lấy console log trên trình duyệt Internet Explorer:**

Link video hướng dẫn: <http://bit.ly/2XPCN9T>

- **Lấy console log trên trình duyệt Edge:**

Link video hướng dẫn: <http://bit.ly/2xF2aws>

- **Lấy console log trên trình duyệt Safari:**

Link video hướng dẫn: <http://bit.ly/30nQUAI>

- **Lấy file log trên điện thoại iPhone**

Một phần mềm để lấy file log trên điện thoại iPhone đó chính là iTools4 (chỉ cần tải về, không cần crack )

Link download: <http://taimienphi.vn/download-itools-7947>



### *Phần mềm iTools 4*

#### **- Lấy file log trên điện thoại Android:**

Đối với các thiết bị android sẽ có nhiều cách để lấy file log khi test app. ( Ví dụ với Samsung thì chúng ta có thể lấy trực tiếp trong thiết bị khi vào phần Developer options) Nhưng dưới đây mình sẽ đề xuất một cách thức để lấy file log chung cho hầu hết thiết bị smartphone Android đó chính là sử dụng Minimal ADB and Fastboot

Cách thức tiến hành như sau:

#### **Các bước set up chuẩn bị:**

1. Cài đặt tool Minimal ADB and Fastboot trên PC: Link download ( nhớ show desktop): <https://drive.google.com/.../1I29fpYUaUJLa8NBvn7c.../view...>

2. Để chắc chắn việc kết nối android và PC tốt nhất thì nên cài đặt driver của loại smartphone cho PC theo link sau: <https://developer.android.com/studio/run/oem-usb#Win10>
3. Vào cài đặt điện thoại bật chế độ USB debug ( trong phần Developer Options )

#### **Tiến hành lấy log file:**

1. Kết nối PC và điện thoại android (đảm bảo việc kết nối tốt như cáp, driver..)
2. Chạy tool ADB sau đó nhập dòng chữ "adb logcat "

(khi đó tool file dòng log sẽ chạy trên màn hình, chúng ta tiến hành tái hiện bug trên điện thoại )

3. Sau khi bug xảy ra chúng ta tiến hành copy toàn bộ dòng log và paste vào 1 file .txt đã tạo sẵn.

Như vậy là chúng ta đã có 1 file log hoàn chỉnh để upload lên Test IO, uTest

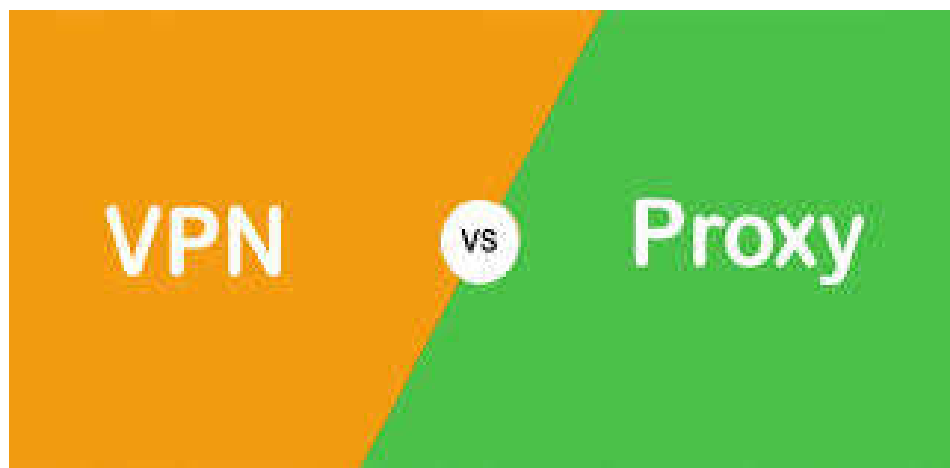
Xem chi tiết tại đây: <https://testertudo.com/cach-lay-log-file-khi-test-app-tren-smartphone-ios-va-android/>

#### 4. Thay đổi proxy và VPN trên thiết bị test.

Một số test cycle khách hàng yêu cầu tester đổi proxy và địa chỉ IP khi tham gia test sản phẩm của họ do vậy tester cần nắm được cách thay đổi proxy trên các thiết bị PC, iPhone, Android.

Mọi người có thể đọc tham khảo file dưới đây mô tả khá chi tiết các bước để thay đổi proxy và VPN của thiết bị test.

Link: <http://bit.ly/ProxyandVPN>



*Thay đổi Proxy và VPN cho thiết bị*

Ngoài ra để phục vụ cho công việc viết bug report các bạn có thể sử dụng các tiện ích được cài trực tiếp trên trình duyệt như **Grammarly** để check ngữ pháp chính tả tiếng anh giúp các bạn tránh bị rejected với những lỗi không đáng có đó.

## Giới Thiệu Các Nền Tảng Test Và Cách Tham Gia Giúp Bạn Kiếm Được 200~1000\$/Tháng

Đây chắc chắn sẽ là phần mà đa phần các tester muốn biết phải không 😊

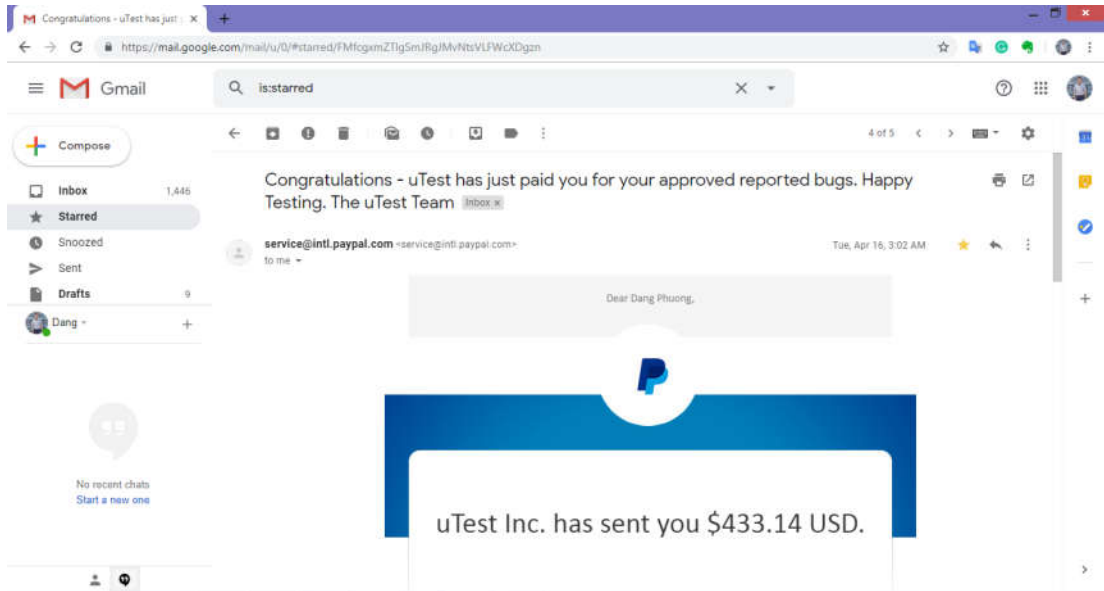
Với bản thân mình qua nhiều năm chiến mảng testing online này và cũng đã kiếm được khá nhiều tiền từ nhiều nền tảng khác nhau thì dưới đây là các nền tảng testing online mà mình đang tham gia và sẽ mang lại nguồn thu nhập khá hấp dẫn cho các tester.



test IO GmbH sent you  
€ 354.33 EUR.

Note from test IO GmbH:

*Kiểm 354.33 EUR từ Test IO*



*Kiểm 433.14\$ từ uTest*

## 1. TEST IO

Là một nền tảng test chắc không còn gì xa lạ đối với mọi người đặc biệt là các thành viên trong group Facebook Cộng đồng tester tự do.



Đây là một nền tảng test khá tối ưu khá nhiều tester lựa chọn để test.

- **Ưu điểm:**

- Số lượng bài test nhiều
- Giá bug cao, tester có thể kiếm được nguồn thu nhập khá ổn nếu đầu tư về thời gian và công sức.
- Có kho tài liệu về testing khá hữu ích (Test IO Academy)

- **Nhược điểm:**

- Tuy số lượng bài test nhiều nhưng mức cạnh tranh giữa các tester cao.
- Cách làm việc của đội ngũ TTL khá khắt khe
- Khả năng bị ban tài khoản cao vì vi phạm các rule của Test IO
- Không được Test IO chấp nhận tạo 2 tài khoản.

- **Các bước kiếm tiền trên nền tảng Test IO:**

Chi tiết mọi người xem tại bài viết sau: <https://testertudo.com/quy-trinh-trinh-kiem-tien-testing-online-tu-nen-tang-test-io/>

## 2. UTEST ( <https://www.utest.com/> )

Là một nền tảng test chắc vẫn còn nhiều người chưa biết tới nhưng theo quan điểm cá nhân mình đánh giá so với Test IO thì uTest không hề thua kém thậm chí ở một số điểm còn vượt Test IO.



### • Ưu điểm:

- Số lượng bài test tương đối ổn
- Giá bug ở mức tốt nhưng so với Test IO thì kém hơn một chút
- Có kho tài liệu về testing khá hữu ích (uTest Academy)
- Có forum để trao đổi tương tác giữa các testers trong cộng đồng

### • Nhược điểm:

- Số lượng dự án cho newbie còn khá ít
- Cách thức tham gia phức tạp
- Để vượt qua các khóa học của uTest academy mất nhiều thời gian

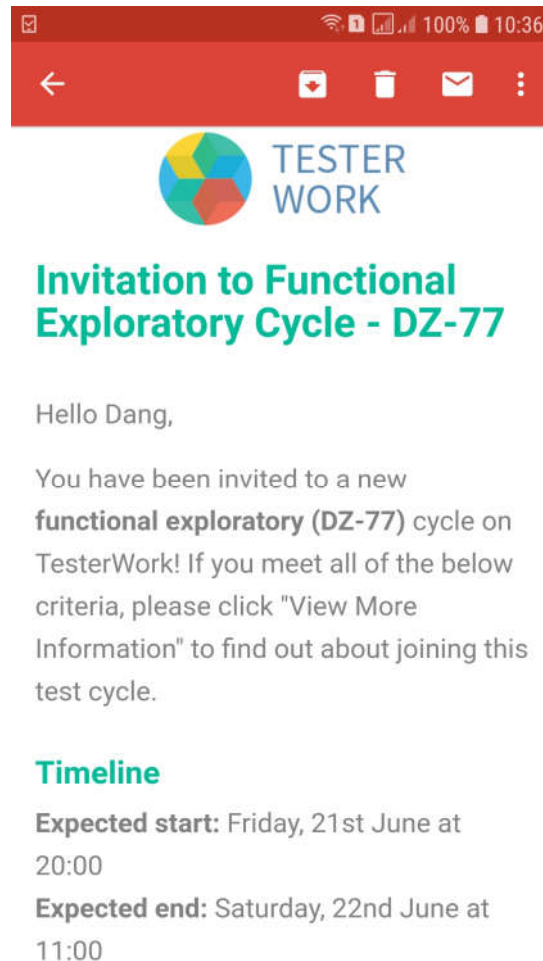
### • Cách thức tham gia:

Mọi người đọc bài viết này để hiểu được về quy trình kiếm tiền online trên nền tảng uTest: <https://testertudo.com/kiem-tien-testing-online-utest/>

Ngoài hai nền tảng test chính trên thì hiện mình cũng tham gia kiếm tiền trên các nền tảng khác nhưng không đủ thời gian để làm hết được, dưới đây mình sẽ liệt kê để các bạn biết các nền tảng khác có thể thử sức.

### 3. TESTER WORK (<http://www.testerwork.com/>)

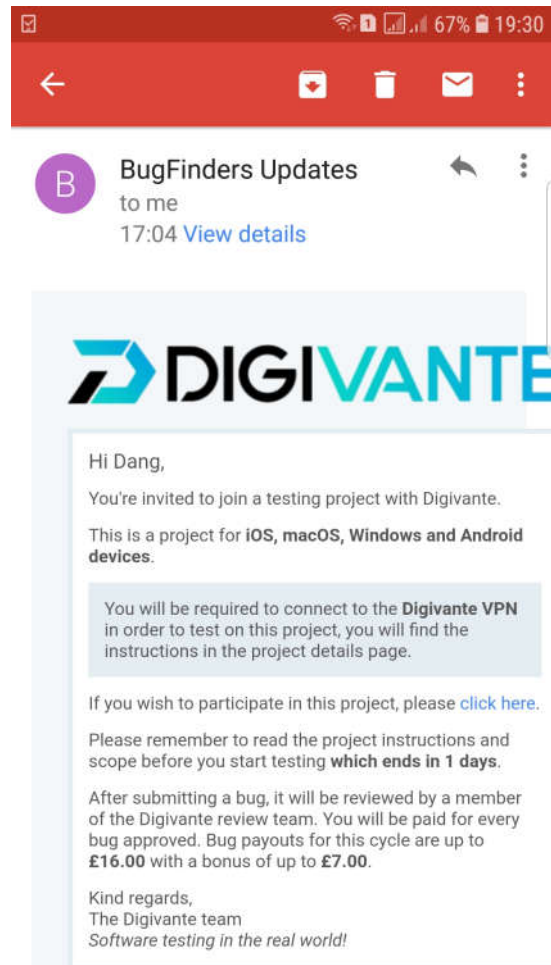
Có khá nhiều dự án nhưng nhược điểm là rất nhiều tester tham gia nên test cycle nhanh full, số lượng bug submit nhanh. Mọi người tham bằng cách đăng kí tài khoản trên tester work, điền đầy đủ các thông tin yêu cầu tương tự như các nền tảng test khác. Sau khi vượt qua bài Quiz test thì công ty sẽ gửi các paid test cycle cho các tester.



*Email mời tham gia test cycle (Tester work)*

- Thêm một vài site nữa để các bạn có thể khám phá mà mình đã kiểm định độ tin tưởng.

- Digivante



*Email mời tham gia test cycle (Bug Finders)*

- Testbirds

- 99tests

# Các Điều Quan Trọng Giúp Bạn Bắt Bug Nhanh Hơn Và Tránh Xa Các Bug Bị Rejected

Khi tham gia test cycle điều mà hầu hết các tester quan tâm sẽ là làm cách nào để bắt bug nhanh và hợp lệ, tránh các bug bị rejected.

## 1. Làm cách nào để bắt bug nhanh hơn chất lượng hơn.

Đây chính là điều mà tester đều muốn thực hiện nó, một số vấn đề mình muốn mọi người chú ý tới nó.

### • Luôn chấp nhận bài test và tham gia nó nhanh nhất có thể:

Việc chấp nhận và tham gia test ngay sau khi nhận được bài test giúp tăng khả năng tìm ra được bug và chất lượng của bug tìm thấy. Do vậy mọi người cần chú ý tới các điểm sau:

- Sử dụng 1 email hoàn toàn mới để đăng kí trên các nền tảng test: điều này giúp tester tránh khỏi các email ngoài phạm vi testing, email rác...
- Kiểm tra lại cài đặt thông báo trên các ứng dụng Gmail, chỉnh sửa chuông thông báo email khác biệt (To dài như báo thức có thể giúp gọi bạn dậy test cả những lúc bạn đang ngủ)

### • Luôn đọc cẩn thận mục Overview và Scope của test cycle:

Đây là một điều rất quan trọng mà đôi khi các tester thường xem nhẹ.

- Mục Overview: tổng quan về test cycle, ở đó tester có thể tìm thấy các thông tin về sản phẩm test, có thể là các cài đặt yêu cầu đặc biệt khi tester tham gia test (setup instruction). Tất cả các điều đó giúp tester tiếp cận tới sản phẩm test nhanh và chuẩn xác hơn
- Mục Scope: Là mục mà khách hàng muốn các tester tập trung test vào phần đấy và chắc chắn rằng tỉ lệ các bug có giá trị cao luôn nằm trong các phần đấy mặc

dù tester có thể tìm thấy bug hợp lệ ở các phần khác nhưng các bug đấy thường chỉ được approved ở mức low. Do vậy để tránh mất thời gian vào các bug low thì hãy tập trung vào mục scope rồi sau đó mới mở rộng thêm khu vực test.

- **Hãy tham gia test như một người sử dụng sản phẩm thông thường:**

Hẳn nhiều tester từng rơi vào trường hợp khi tham gia một test cycle và tìm mãi không ra bug dẫn đến việc nóng vội, chán nản.

Theo mình khi bước vào test một sản phẩm thì hãy luôn giữ trạng thái test thoải mái và hãy xem là mình đang sử dụng sản phẩm đó chứ không phải là đang “test” nó. Việc này giúp tester có thể duy trì thời gian testing kéo dài và không bị stress điều mà bất kì một người nào khi làm việc với máy tính và điện thoại trong thời gian dài hay mắc phải.

- **Tim lỗi theo kinh nghiệm:**

Sau khi tham gia một vài cycle thì bạn có thể nhận ra rằng rất nhiều website, ứng dụng đều có các chứng năng tương tự mà ở đó các bug thường hay xuất hiện. Đây chính là câu trả lời cho câu hỏi tại sao mình lại bắt bug chậm hơn các tester khác? Và dĩ nhiên kinh nghiệm không phải là điều mà tester có thể đạt được một sớm một chiều, đó là cả một quá trình trải qua rất nhiều các test cycle, log hàng trăm, hàng nghìn, thậm chí hàng chục nghìn bug, tất cả điều đó luôn giúp các tester có kinh nghiệm khi bước vào 1 test cycle thì họ biết họ cần test phần nào, các mục nào cần tập trung test mà theo họ ở đó chắc chắn sẽ xuất hiện bug.

Hay nói đơn giản hơn với mỗi loại sản phẩm test thì các tester luôn định hình sẵn trong đầu một loạt các testing list cho loại sản phẩm đó.

Dưới đây là một vài suy nghĩ về cách tìm bug của cá nhân mình như sau:

- Giữa 1 bài test mà có SP và PC thì ưu tiên smartphone trước. Còn nếu cho test IE thì ưu tiên PC -> Vì nếu test trên SP thì mình có thể test đc trên Safari và mình cũng có thể check function như trên PC, đồng thời có thể bonus thêm Visual hoặc Visual->Function, đa số mấy trang website thường làm kiểu desktop-first, nên qua SP dễ bug lắm.

- Ưu tiên check các feature mới update trước (nó được nhắc đến trong mục Goal hoặc gần đó) có thể nó chỉ đc thấp hoặc cao nhưng để có bug -> Sau khi update hoặc fix bug, sẽ sinh ra hiện tượng để thêm bug.
- Cá nhân mình PDP (Product detail page) và PLP (Product list page) thì check kỹ, nó hay lỗi filter, sort, với mấy cái vớ vẩn về thông tin product.
- Chỗ nào có Sale thì check mấy trang Sale trước (PLP) thường xảy ra bug sort, với cart khi đi cùng sản phẩm sale thường xảy bug giá tiền, promotion ở cart.
- Log bug chậm -> cái này thì tải bộ gõ 10 phím về cải thiện gõ bàn phím vậy :)))))) và phán định đó là bug hay không rồi log tốc độ thôi, còn việc phán định nó có là bug hay không, nhanh hay chậm là tùy vào việc đã đọc kỹ mô tả bài test hay chưa, chứ tới lúc tìm ra bug, lại quay lại xem thử cái bug đó có OOS hay không, thì có khi người ta submit vài giây trước.
- Nghĩ là bug nhưng không phải là bug -> trước bắt đầu bài test hãy cố gắng dành 10-20 phút để đọc nhớ tất cả trường hợp in scope và out scope thôi, chứ ngồi suy nghĩ làm gì, xuất phát chậm nhưng mà chắc.
- Làm sao để bắt bug hiệu quả -> cá nhân mình là developer nên trước khi test một function của 1 feature nào đó, thì mình sẽ nghĩ tất cả các case có thể xảy ra, case dùng để test tính liên quan giữa các function trong feature, ví dụ: sort + filter + paging; chọn màu + hình ảnh sản phẩm lúc bình thường và sau khi phóng to; upload hình sau đó thử delete hình đó. Feature với feature, ví dụ như: bạn follow 1 user thì thông tin có đc update ngay lúc đó ko, ở màn hình khác có update thông tin đó ko, bạn thêm mới một user thì user kia có thể tìm thấy user đó ko, etc. Nói chung là bạn nên tạo ra kịch bản trước khi muốn test feature nào đó, để bạn có hướng đi trước, còn hơn là test kiểu mò mẫm bug. Và đương nhiên là bạn làm nhiều bạn sẽ nghĩ ra càng nhiều case, thậm chí chả cần nghĩ, chỉ cần phán đoán chỗ đó có bug là chắc chắn có bug, cái này thì gọi là đẳng cấp thượng thừa, 1 tiến tấn thăng thành tiên :)). Và mình là Tiên, nhưng là tên bố mẹ đặt thôi ạ.
- Know bug, dup, search -> Đừng tin tưởng vào thanh search by key word, bởi vì tay trái bạn đếm tiền tay phải bạn còn phải đếm lại; con tim thì cho đó là bug nhưng con tim có bao giờ nằm bên phải đâu, vì vậy bạn nên:
  - o filter tất cả các bug theo feature bạn đang muốn test
  - o Đọc lướt qua title của bug, ngẫu nhiên đọc tiếng anh, yêu nước thì chuyển qua tiếng Việt đọc cho dễ.
  - o Nghi ngờ con bug nào trùng thì vô coi clip của họ tí, chứ đôi khi nó trùng thật.
  - o Nếu vẫn tin tưởng search by key work thì hãy search bằng các label trên cái web bạn đang test dành riêng cho element, ví dụ: First name, Last name, Waiter.... Bởi vì search của testio là nó search all content of bug.
  - o Khi bạn đang log bug sẽ có các bug gợi ý bên cạnh, thử đọc xem có trùng không, đôi khi mình tìm ko ra mà nó lại gợi ý đúng á. Vì mấy ông log bug mô tả xí lắc léo lắm.

- **Quy trình xử lý:**

Mục đích chính của tất cả các web bán hàng sẽ là tập trung vào bán sản phẩm. Nên các bước xử lý đơn hàng và tất cả các chức năng liên quan tới sản phẩm đều được xem là quan trọng. Nếu bạn có thể tìm được một lỗi trong thanh toán đơn hàng thì những bug đó mặc định sẽ là lỗi Critical có mức ưu tiên cao nhất. Không có nghĩa là bug sẽ chỉ nằm ở đó, và bạn cứ cắm vào đó mà bỏ qua những nơi khác. Tiền to không tiền được liền, thì tiền nhỏ bạn cũng đừng bỏ qua mà thử ở chức năng khác nhé

- **Danh mục phân loại sản phẩm:**

Đây sẽ là nơi để bạn săn bugs, gồm những trang mô tả chi tiết sản phẩm, những điều hướng chính, chức năng tìm kiếm – search, sắp xếp kết quả tìm kiếm – Sort, trang kết quả tìm kiếm – Paging (nếu nó nằm trong phạm vi đề bài giao cho bạn). Dĩ nhiên là bạn sẽ vào vai một khách mua hàng, bạn không cần phải đăng kí để làm điều đó

- **Nơi bạn săn bugs:**

Tự hỏi bản thân, tính năng nào là chính và số lượng những lựa chọn/ chức năng để xác định số lượng test case cho từng mục. Rà soát theo 1 cái checklist để không bỏ sót tất cả những góc ngách nơi mà những con bugs ẩn náu. Những nơi bạn cho là mà khả năng cao có bugs? Ví dụ: trên trang mô tả chi tiết sản phẩm lỗi có thể là màu sắc, đơn vị kích thước, bảng tra cứu hay hướng dẫn khách chọn kích thước phù hợp, ... thử lọc, tìm, sắp xếp, ect

- **Những góc ngách khác:**

Đừng quên những góc ngách khác, có thể là mục đăng kí tin mới – Newsletters, Header, Footer, trang Landing, chức năng chia sẻ Facebook, Twitter, Instagrams ...ect

- **Thêm xíu gợi ý nữa nè:**

Cố gắng action rồi trông đợi một kết quả, nhưng thực tế lại khác → đó có thể là bug đó nhé.

Cố gắng nhập vai khách hàng càng giống càng tốt: Người cố gắng tìm hàng họ đang cần mua, nhưng sản phẩm họ đang quan tâm

Cài đặt thêm vài browsers trên devices của bạn, có thể bug xảy ra ở một số browser không là tất cả, với một số phiên bản của browser cũng là một gợi ý

Tiến hành báo cáo bugs cho team leader, tiếp nhận phản hồi để cập nhật và hoàn thiện báo cáo bugs.

Nếu bạn nhận một thông báo hoàn thiện báo cáo bugs, bạn có 18 giờ để cập nhật và gửi báo cáo mới, nếu bạn không làm được, bugs sẽ tự động bị reject.

Chất lượng từ khóa bạn dùng, một báo cáo chất lượng cũng giúp bạn tăng khả năng được xem xét nhanh chóng hơn. Team leader không mấy quan tâm đến những báo cáo dài dòng lan man.

Đừng báo cáo lỗi ở mức độ nghiêm trọng một cách bừa bãi. Ngoài việc tìm bugs, đánh giá mức độ bugs đúng cũng được xem là công việc của một tester

Nếu bạn không thể tìm được bug ngay bước đầu tiên, đừng bỏ cuộc, hay tìm thêm cho mình những gợi ý khác trong 'Known Bugs'. Nhưng lưu ý những bugs này sẽ không được báo cáo lại.

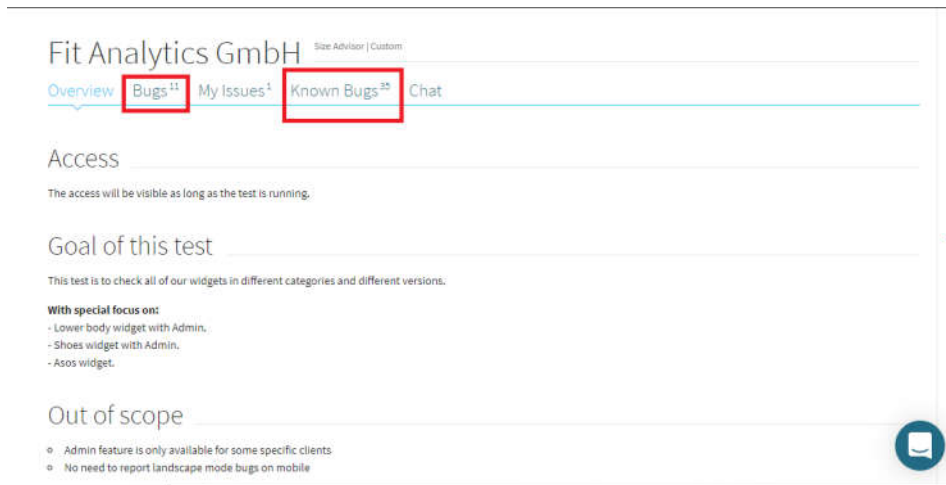
Những lỗi nhỏ trong chức năng sẽ tìm thấy trong những hyperlink ở footer, header, sharing ...

Cố gắng hệ thống phương pháp kiểm thử, kiểm thử từng chức năng, kiểm thử tích hợp nhiều chức năng,

Nếu bạn không thể tìm thấy bugs trên the first onboard hay cho mình cơ hội khác với onboard khác, với một web khác

**- Học hỏi từ các bug của các tester khác trong cùng 1 cycle:**

Cái này là một mẹo khá hay đối với các tester chưa có kinh nghiệm khi tham gia test cycle.



+ Các bug được accepted trong mục known bugs sẽ giúp các bạn cách viết một bug report chuẩn mà không bị comment của Test leader của test cycle đấy.

+ Các bug được các tester submit trước mình trong mục Bugs list sẽ giúp chúng ta định hướng tập trung test tới các phần đó tránh test ngoài phạm vi yêu cầu.

## 2. Tránh xa bug bị rejected.

Một điều không thể không nhắc tới bên cạnh việc log bug nhanh thì việc tránh xa các bug có thể bị rejected cũng quan trọng không kém.

### • Bug bị rejected bởi những lý do nào?

- **Out of scope (OOS):** Bug được submit ngoài phạm vi test
- **Duplicate:** Bug submit bị trùng với bug đã được báo cáo trước đó hoặc trùng với bug trong list known bugs.
- **Rejected by test leader:** Tester không chỉnh sửa bug theo comment mà test leader yêu cầu. (Đối với test IO thì sau 18h nếu không chỉnh sửa bug theo comment thì bug đấy sẽ bị rejected)
- **Reject by customer:** Khách hàng không chấp nhận bug mà tester báo cáo do bug này không đạt chất lượng mà khách hàng mong muốn
- **Working as design (WAD):** Các bug mà nó hoạt động đúng như thiết kế hoặc là lỗi nhỏ mà khách hàng đã biết và sẽ không fix.

- **Bug bị rejected có ảnh hưởng gì không?**

Câu trả lời là **CÓ**. Ở các nền tảng test mỗi tester có một profile riêng và được công ty đánh giá quá trình testing của testing qua chỉ số rank. Tester có nhiều bug được approved và ít bug rejected đồng nghĩa với rank cao.

Rank là một trong những yếu tố để công ty lựa chọn tester để mời tham gia các dự án test cycle ngoài các yếu tố khác như: thiết bị test, quốc gia...

- **Làm cách nào để tránh xa bug bị rejected.**

Mọi người chú ý vào các mục sau để giảm tối đa việc bị rejected một cách đáng tiếc như sau:

- Đọc kỹ phần out of scope tại tab overview để tránh test và log bug trong các mục đấy
- Luôn trả lời comment của Test leader sớm nhất có thể.
- Trước khi log bug luôn kiểm tra lại bug và so sánh bug với các bug đã log trước đó và các bug trong mục known bug bằng cách nếu có thể chúng ta nên đọc lướt qua các bug mà các tester khác đã log trước đó hoặc các bug trong mục known bugs.

Không thể loại bỏ hoàn toàn rejected bug nhưng trên đây là những cách để tester có thể tránh bug bị rejected một cách tối đa.

- **Có nên dispute bug.**

Dispute bug hay hiểu đó là kháng cáo bug bị rejected. Đây là một cách để các tester có thể bảo vệ bug mà mình report khi bug từ chối. Khi dispute bug thành công tester sẽ nhận được tiền bonus cho bug đấy và đi đôi với nó khi dispute bị fail thì sẽ có nhiều tác động tiêu cực đến tester. (Có thể bị block chức năng dispute trong một thời gian, chỉ số testing rank bị tác động xấu.) Do vậy luôn cân nhắc trước khi tiến hành dispute.

## 6 Điều Hữu Ích Giúp Bạn Thành Công Trong Công Việc Testing Online

Dưới đây mình muốn trích dẫn một bài viết mà mình đọc được từ một test manager bên mảng testing online chia sẻ về kinh nghiệm thành công khi làm việc với testing online.

### 1. HÃY CÓ TRÁCH NHIỆM VÀ TRUNG THỰC.

- Tốt nhất là chỉ chấp nhận bài test khi bạn có thể bắt đầu test càng sớm càng tốt.

Điều này giúp các bạn có nhiều cơ hội hơn tiếp cận với các bug.

- Đừng bao giờ chấp nhận nhiều bài test hơn việc bạn có khả năng tham gia.

Hãy dành thời gian tập trung vào một test cycle trước khi nghĩ tới các cycle khác, việc nhận quá nhiều cycle so với khả năng tham gia làm cho chúng ta bị “loãng” khiến kết quả đạt được không cao.

- Hãy đọc kỹ mục overview (2 lần nếu bạn có thể) trước khi chấp nhận nó.

Nếu bạn không thể hiểu được yêu cầu của bài test hãy kiểm tra mục thiết bị test, ngôn ngữ test hoặc có thể sử dụng Google tìm kiếm về sản phẩm test để hiểu nó là gì.

- Nếu bạn có một vấn đề riêng liên quan tới việc tham gia thực hiện test trên cycle thì hãy liên lạc mô tả tình huống đó cho test leader về vấn đề đấy trên mục Chat.
- **ĐỪNG** bao giờ báo cáo lỗi với tư cách là một người giữ chỗ (Place holder) mà không có full tập đính kèm hoặc các thông tin cần thiết cho chu kì để thay đổi chúng sau đó.

Điều này khiến bạn trở thành một người không trung thực và vi phạm quy tắc ứng xử. Các tester khác tham gia trong test cycle hoàn toàn có thể tố cáo hành

vi gian lận này cho test leader và bản thân test leader có thể kiểm tra lịch sử chỉnh sửa bug gian lận đấy và đưa ra phán quyết chính xác.

## **2. BÁO CÁO BUG CHẤT LƯỢNG**

- Hãy hoàn thành báo cáo một cách rõ ràng, chính xác và đầy đủ nhất có thể. Hãy kiểm tra lại lỗi chính tả trước khi submit bug đấy. Tập trung vào độ chính xác tiêu đề lỗi (Bug title) vì tiêu đề lỗi tốt sẽ giúp không chỉ tester mà khách hàng có thể lọc các báo cáo bug bị trùng lặp một cách dễ dàng hơn.
- Hãy chuẩn bị kĩ càng thiết bị test trước khi tham gia test cycle. Tắt tất cả các thông báo trên thiết bị di động điều mà gây ra cản trở khi bạn đang ghi lại lỗi hoặc nếu trên máy tính thì nên tắt các cửa sổ không cần thiết và nên bảo mật các thông tin cá nhân. Một điều quan trọng nữa đó là hãy loại bỏ các âm thanh tạp không liên quan đến bug khi bạn ghi lại lỗi vì điều này có thể gây khó chịu cho người kiểm tra lỗi.
- Tập trung vào chất lượng hơn số lượng và luôn cố gắng cải thiện hiệu suất làm việc. Nếu bạn tham gia nhiều hơn một lần cho cùng một sản phẩm, hãy thử tiếp cận sản phẩm test theo một cách khác so với lần trước để vượt qua các khu vực test mới các bug mới hoặc cố gắng tái hiện lại các bug mà các tester đã báo cáo như một cách để nhận được bonus hoặc rút ra thêm được nhiều kinh nghiệm khác.

## **3. HÃY LUÔN THỂ HIỆN LÀ NGƯỜI THÂN THIỆN, LỊCH SỰ VÀ CÓ HỌC THỨC.**

- Hãy luôn sử dụng các từ "Please", "Thank you" khi bạn muốn hỏi một điều gì đó trong mục Chat. Đừng bao giờ tỏ thái độ thô lỗ hoặc hỏi quá nhiều về cùng một câu hỏi khi bạn chưa nhận được phản hồi. Bạn cần phải kiên nhẫn vì chúng ta đang làm việc trên các nền tảng quốc tế và chúng ta đang sống ở những múi giờ khác nhau và các leader của chúng ta luôn xứng đáng được nghỉ ngơi.

#### 4. HÃY TRỞ THÀNH TESTER MÀ MỌI NGƯỜI ĐỀU MUỐN LÀM VIỆC CÙNG.

- Khi tham gia một test cycle mới, hãy tưởng tượng sản phẩm đó là của bạn và bạn đang làm việc trong một văn phòng với các thành viên trong nhóm. Nếu bạn thấy một lỗi OOS hoặc Duplicate thì hãy cố gửi một tin nhắn riêng cho tester đó để giúp họ loại bỏ bug đấy. Tôi chắc chắn rằng bạn cũng muốn nhận được những tin nhắn này nếu rơi vào tình huống tương tự phải không.
- Hãy đối xử tốt với các tester khác, trong mỗi cycle không phải để so sánh ai giỏi hơn ai mà hãy nghĩ rằng chúng ta đang làm việc trong cùng một nhóm không phải là đối thủ!
- Hãy khiêm tốn và chấp nhận nếu chúng ta mắc lỗi. tất cả chúng ta cũng là con người chứ không phải robot do vậy việc mắc sai lầm là điều bình thường, mỗi sai lầm sẽ là một bài học kinh nghiệm.

#### 5. HÃY LÀ MỘT TESTER CÓ TỔ CHỨC!

*Cách làm việc có tổ chức sẽ giúp ích bạn rất nhiều cho công việc testing đặc biệt là khi bạn làm việc với số lượng test cycle lớn.*

- Tạo 2 folder trong đó folder "Old Cycle" sẽ chứa các file trong bug report của các bài test đã đóng, một folder " Current Cycle" chứa các file của cycle đang hoạt động. Ở mỗi cycle sẽ phân chia ra các bug nhỏ điều này giúp bạn quản lý bug tốt hơn và hãy thực sự xóa nó khi bug được accepted nếu cần.
- Luôn chuẩn bị một cuốn Notebook bên cạnh khi tham gia test, ở đây bạn có thể ghi các thông tin quan trọng của cycle đấy ra hoặc các ý tưởng test nảy sinh ra để thực hiện test cho sản phẩm hay đơn giản ghi cách thay đổi proxy để khi cần bạn chỉ cần mở ra xem mà không cần xem lại video.

## 6. LUÔN LÀ MỘT THANH VIÊN NĂNG ĐỘNG TÍCH CỰC GIAO TIẾP VỚI CỘNG ĐỒNG.

- Các nền tảng test đều có các cộng đồng hoặc là các cộng đồng tester trên Facebook, blog (Ví dụ : Group Facebook : Cộng đồng Freelance tester) và việc bạn tích cực tham gia giao lưu trên đó sẽ mang lại cho bạn nhiều thông tin hữu ích.
- Bạn sẽ nhận được những câu trả lời cho những khó khăn khi làm việc với các cycle
- Cập nhật những update về quy định mới mà các nền tảng đưa ra.
- Nhận được nhiều động lực trong công việc đến từ các tester khác.

Mình tin rằng việc duy trì các điều trên cùng với sự trau dồi về kiến thức thì không có lý do gì mà các không thành công trong công việc testing online.

Hãy luôn giữ được đam mê khi làm việc online.

Happy Testing!

# TỔNG KẾT

Cảm ơn mọi người đã đọc hết cuốn Ebook này. Mình hy vọng các kiến thức ở trên sẽ là tiền đề để các bạn có thể bắt tay tham gia mảng kiếm tiền testing online và kiếm thêm cho bản thân khoản thu nhập thêm trong khoảng thời gian free-time

## Tổng hợp các kiến thức cần chú ý:

1. Khái quát quá trình kiếm tiền trên nền tảng Test IO:  
<https://testertudo.com/quy-trinh-trinh-kiem-tien-testing-online-tu-nen-tang-test-io/>
2. Khái quát quá trình kiếm tiền trên nền tảng uTest:  
<https://testertudo.com/kiem-tien-testing-online-utest/>
3. Cách tạo tài khoản Payoneer ( nhận bonus \$50 khi nhận đủ về tài khoản \$1000): <https://testertudo.com/dang-ky-payoneer/>
4. Tổng hợp bug mẫu tham khảo: <https://testertudo.com/bug-list/>

## • Thông tin liên hệ:

- 👉 FB Đặng Phương: <https://www.facebook.com/bwin7cr>
- 👉 Testertudo team: <https://www.facebook.com/testertudo>
- 👉 Facebook group: <https://www.facebook.com/groups/freelance.Tester>
- 👉 Website: <https://testertudo.com/>