

# User Story

Tags

Definition

User Story là gì?

Tại sao cần có user story?

Mẫu user story và ví dụ

## Definition



Nói ngắn gọn thì “user story” **là một phương tiện giải thích không chính thức và tổng quan về một tính năng của phần mềm**, nó được viết từ góc nhìn của người dùng cuối. Mục đích của “user story” là diễn đạt cách một tính năng phần mềm sẽ mang lại giá trị cho người dùng.

Có lẽ nhiều người nghĩ rằng user stories (số nhiều của user story) đơn giản là yêu cầu hệ thống phần mềm. Nhưng thực tế, chúng không phải như vậy. Một yếu tố quan trọng của phát triển phần mềm theo phong cách linh hoạt (Agile) là đặt con người lên hàng đầu và user story đặt người dùng cuối (end users) ở trung tâm của cuộc trò chuyện.

Những câu chuyện này sử dụng ngôn ngữ tự nhiên (như tiếng Anh, Việt) để cung cấp ngữ cảnh cho nhóm phát triển phần mềm. Sau khi đọc một user story, nhóm sẽ biết rõ tại sao họ đang phát triển, họ đang xây dựng cái gì và giá trị mà tính năng hay phần mềm này mang lại là gì.

User story là một trong những thành phần cốt lõi của phương pháp phát triển phần mềm theo hướng Agile. Chúng giúp tạo ra cách làm việc tập trung vào người dùng mỗi ngày và thúc đẩy sự hợp tác, sáng tạo và tạo ra một sản phẩm tốt hơn tổng thể.

## User Story là gì?

User story là đơn vị công việc nhỏ nhất trong trong dự án Agile. Là một mục tiêu cuối cùng, không phải là một tính năng, được diễn đạt từ góc nhìn của người sử dụng phần mềm.

Một user story là một giải thích không chính thức, tổng quan về một tính năng phần mềm, được viết từ góc nhìn của người dùng cuối hoặc khách hàng. Mục đích của user story là diễn đạt cách một công việc sẽ mang lại giá trị cụ thể cho khách hàng. Lưu ý rằng “khách hàng” không nhất thiết phải là người sử dụng cuối. Ví dụ như công ty A thuê chúng ta làm phần mềm cho họ, thì người dùng cuối có thể là khách hàng của công ty A, hoặc còn có thể là những khách hàng hoặc đồng nghiệp nội bộ trong công ty của bạn như một ngân hàng, thì nhóm nhân viên nghiệp vụ ngân hàng sẽ là “khách hàng” của nhóm IT làm ra hệ thống phần mềm cho ngân hàng đó.

User stories chỉ là một vài câu đơn giản bằng ngôn ngữ tự nhiên dễ hiểu, mô tả kết quả mong muốn, không mô tả chi tiết vấn đề – các thông tin này sẽ được thêm vào sau khi được sự đồng ý của cả nhóm.

Để phát triển một chức năng hoàn thiện sẽ cần có nhiều user story. Lúc này các nhóm phát triển phần mềm thường sử dụng epic để tập hợp các user story liên quan lại với nhau.

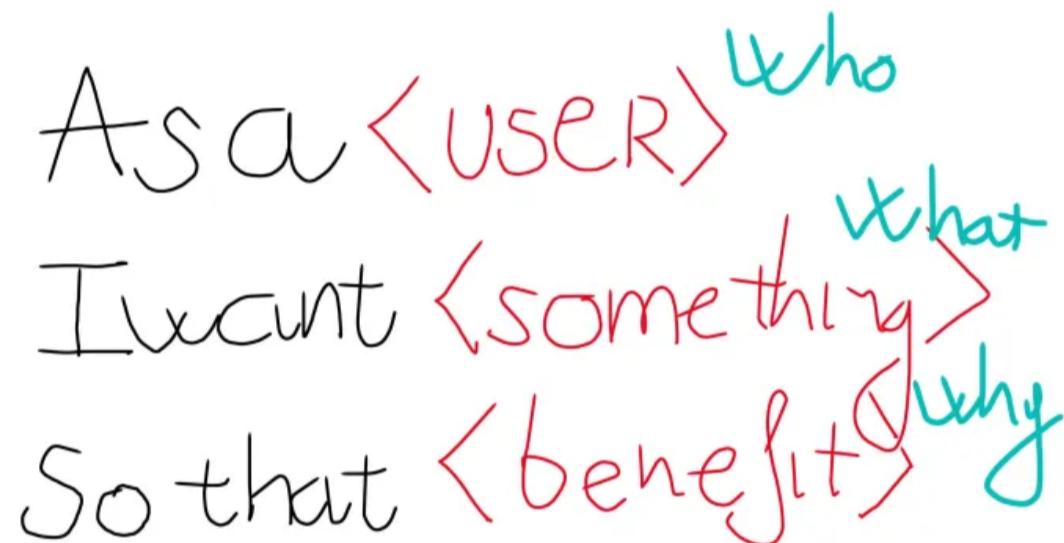
## Tại sao cần có user story?

Đối với các nhóm phát triển mới làm quen với Agile thì user story có vẻ như là một bước không cần thiết. Họ cho rằng, chỉ cần chia dự án lớn (epic) thành một loạt **task** (nhiệm vụ/công việc) và tiến hành lập trình thôi. Nhưng user story mang lại cho nhóm thông tin quan trọng và liên kết các task với **giá trị mà các task đó mang lại**.

User story mang lại nhiều lợi ích quan trọng như:

- **Tập trung vào người dùng:** User story giúp nhóm phát triển phần mềm tập trung vào người dùng cuối (end-user: người trực tiếp sử dụng sản phẩm phần mềm được tạo ra). Danh sách task chỉ giúp nhóm tập trung vào nhiệm vụ cần được hoàn thành, trong khi đó danh sách các user story sẽ giúp nhóm tập trung vào **giải quyết vấn đề cho người dùng thực** (real user).
- **Khuyến khích sự hợp tác:** User story tạo điều kiện cho sự hợp tác giữa các bên liên quan. Với mục tiêu đã đề ra, nhóm phát triển phần mềm có thể cùng nhau quyết định cách tốt nhất để phục vụ người dùng và đạt được mục tiêu đó.
- **Thúc đẩy giải pháp sáng tạo:** User story khuyến khích nhóm suy nghĩ quyết đoán (critical thinking) và sáng tạo về cách tốt nhất để **giải quyết mục tiêu chung**.

## Mẫu user story và ví dụ



User story thường được biểu diễn theo một câu đơn giản và có cấu trúc như sau:

"**As a [persona], I [want to], [so that].**"

Trong đó:

- "**As a [persona]**": Chúng ta đang phát triển chức năng này cho ai? Chúng ta không chỉ cần một tên chức vụ, mà còn cần một **bức tranh tổng thể về người đó**. Ví dụ An, thì nhóm chúng ta nên có **một hiểu biết chung** về **An là ai**. Chúng ta phải hiểu cách họ làm việc, suy nghĩ và cảm nhận như thế nào. Chúng ta có sự thấu hiểu và đồng cảm với An. Có như vậy nhóm chúng ta mới phát triển đúng chứng năng mà An (một vai trò trong phần mềm) đang cần.
- "**Wants to**": Ở đây, chúng ta đang mô tả ý định của họ — không phải là các tính năng mà họ sử dụng. Điều họ thực sự đang cố gắng đạt được là gì? Mô tả này không liên quan đến **cách thức triển khai** — nếu bạn mô tả bất kỳ phần nào của giao diện người dùng mà không nói về mục tiêu của người dùng, bạn đã bỏ qua điểm chính.
- "**So that**": Làm thế nào mong đợi tức thì của họ có thể hòa mình vào bức tranh tổng thể? Lợi ích chung mà họ đang cố gắng đạt được là gì? Vấn đề lớn đang cần phải giải quyết là gì?

Ví dụ, một user story như thế này:

- As An, tôi muốn mời bạn bè của tôi, để chúng tôi có thể sử dụng dịch vụ này với nhau
- As Minh, tôi muốn tổ chức công việc của tôi, để tôi có thể cảm thấy mọi thứ trong tầm kiểm soát
- As a người quản lý, tôi muốn hiểu được đồng nghiệp và nhân viên của tôi, để tôi có thể có được các báo cáo hiệu quả hơn về thành công và thất bại của nhóm mình.

Cấu trúc này là **không bắt buộc**, nhóm bạn có thể mô tả user story theo dạng khác, nhưng cấu trúc này rất hữu ích để giúp xác định **định nghĩa đã hoàn thành** (definition of done: như thế nào thì được gọi là đã xong). Khi persona (tính cách, vai trò trong hệ thống) đó có thể nhận được giá trị mong muốn của họ, thì "câu chuyện" được xem là **đã hoàn thành**. Các nhóm phát triển phần mềm có thể tự **xác định cấu trúc riêng** của mình, nhưng sau đó phải tuân thủ nó.

