# CSS Thuần

## Hiển thị … khi text bị tràn ra màn hình

text-overflow: clip|ellipsis|string|initial|inherit;

* **clip**: là giá trị mặc định, nó sẽ kẹp các văn bản.
* **ellipsis** : thêm ba dấu chấm (...) nếu text bị tràn ra ngoài
* **string** : tự định nghĩa đoạn text nào đó thêm vào khi bị tràn ra ngoài.
* **initial** : thiết lập giá trị mặc định
* **inherit**: kế thừa giá trị từ thẻ HTML cha.

Có một số giá trị sẽ không chạy trên trình duyệt Chrome, Opera, ... nên bạn phải cân nhắc khi sử dụng. Tuy nhiên thuộc tính ellipsis thì hoạt động hầu hết các trình duyệt mới hiện nay.

## Text xuống dòng hay không

white-space: normal; 🡺 xuống dòng bình thường (mặc định thì sẽ có chế độ này)

white-space: nowrap; 🡺 không xuống dòng

## Calm (hàm trừ lấy độ dài)

Dùng để tính toán trừ ra độ dài của một div

width: calc(100% - 100px);

## invisible

class invisible trong boostrap : chỉ không cho hiển thị nhưng vẫn giữ chỗ

display: none : không cho hiển thị và không giữ chỗ

# core-ui

## Build core-ui

Tải thư mục về giải nén thì tại project coreui (có file package)

Mở cmd lên và chạy lệnh: npm install 🡺 lúc này xuất thư mục node\_module

Tạo thư mục dist: npm run build 🡺 lúc này thư mục dist sẽ xuất hiện

Dùng npm run để thấy hết tất cả các script

# boostrap

## Class

Tham khảo:

<https://getbootstrap.com/docs/4.2/utilities/borders/>

### class hay dùng

.table-hover // BẢNG HOVER

.text-danger // thay màu chữ

.https://getbootstrap.com/docs/4.1/utilities/display/ // display

.animated .fadeIn // hiệu ứng hiện element ra một cách mượt mà

## Thuộc tính data-

Phải CHÚ Ý rằng mọi plugins của Bootstrap có thể áp dụng chỉ đơn thuần thông qua HTML, không cần viết thêm dù chỉ 1 dòng code JavaScript. Đây gọi là API thứ tự đầu tiên của Bootstrap (Bootstrap’s first-class API) và bạn phải xem đây là giải pháp trước nhất khi làm việc với plugins.

Ví dụ, chúng ta có thể kích hoạt 1 modal, không dùng code JavaScript, như sau, thiết lập thuộc tính data-toggle="modal" cho đối tượng kích hoạt như button hay liên kết <a></a>, data-target=#myModal là thiết lập modal có id là myModal sẽ được kích hoạt/hiển thị, data-backdrop=”static” là không đóng modal khi user clicks ra ngoài modal, và data-keyboard=”false”là không đóng modal khi user nhấn phím escape. 4 thuộc tính được bỏ trong 1 tag HTML kích hoạt duy nhất:

<button type="button" data-toggle="modal" data-target="#myModal" data-backdrop="static" data-keyboard="false">Launch my modal</button>

### Danh sách các thuộc tính data-

Link tham khảo:

<https://getbootstrap.com/docs/3.4/javascript/>

npm

## Modals – popup

### Truy nhập modal

Hãy thêm role="dialog" vào .modal, thuộc tính aria-labelledby="myModalLabel" dùng để tham chiếu đến phần tiêu đề của modal, và aria-hidden="true" để các công nghệ hỗ trợ bỏ qua các phần tử DOM của modal.

Thêm vào đó, bạn cũng có thể thêm phần mô tả cho modal của bạn thông qua thuộc tính aria-describedby trên .modal.

## Vùng chứa – Container

.container : cung cấp vùng chứa có độ rộng cố định

.container-fluid : cung cấp độ rộng toàn phần, tùy theo kích thước màn hình.

# Sass

## Link

**SASS**

<https://viblo.asia/p/hoc-sass-trong-15-phut-wjAM7y0LvmWe>

<https://thachpham.com/web-development/html-css/huong-dan-su-dung-sass-de-viet-css.html>

**GULP**

[**https://gulpjs.com/**](https://gulpjs.com/)

<https://github.com/PhuongNamCorpsIntern/workspace/issues/12>

<https://viblo.asia/p/gulp-va-nhung-khai-niem-co-ban-nhat-oOVlYyjnl8W>

**Phần mềm để chạy sass**: koala (tải về và chạy)

Sử dụng Output (Compresses) để nén file lại

## Quy tắc xếp chồng



Sass Kq

Cách lấy giá trị trước đó:





& tương ứng chính là thẻ li kq

## Sử dụng biến:

Lưu giá trị thường xuyên sử dụng lại

🡺 

## Quy tắc Mixin

🡺 

### Truyền tham số vào mixin:

🡺 

🡺 

## Extends

🡺 

Chỉ kế thừa không in ra:

🡺 

## Các mệnh đề điều kiện:



Tạo file scss:

- Tạo một file tên : \_typography.scss

File này sẽ không hiển thị trong thư mục. Dùng lệnh  để import file này vào file scss

## Framework cho sass

Bourbon (nên xem)

Bourbon NEAT : chia cột (nên xem)

bitters

Refils

## Thực hành: Dùng sass và koala

- Viết code trong sublime text

# GULP

[**https://gulpjs.com/**](https://gulpjs.com/)

<https://github.com/PhuongNamCorpsIntern/workspace/issues/12>

<https://viblo.asia/p/gulp-va-nhung-khai-niem-co-ban-nhat-oOVlYyjnl8W>

# Yarn

Tất cả các gói được sử dụng sẽ được mô tả trong file package.json nằm trong thư mục gốc của dự án.

Sử dụng npm để quản lý gói phần mềm có thể dẫn đến trường hợp hai máy có cùng một file cấu hình package.json nhưng lại có các phiên bản của các gói khác nhau và nảy sinh các lỗi “bug on my machine”

# Cài đặt phần mềm tiện dụng

Bootlink: Kiểm tra cấu trúc HTML

<https://github.com/twbs/bootlint>

# Jquery:

## Kiểm tra checkbox:

$("#check\_semrush\_project\_id").is(":checked") 🡺 kiểm tra có check hay không

## Input

$("#input ").prop('disabled', true); 🡺 bật disable cho thẻ input

$("#input ").prop('disabled',false); 🡺 tắt disable cho thẻ input

## Function

### Gọi chính nó trong function click.

* *HTML*:

<span class="input-group-text" **onclick="del\_properties(this);"**><i class="fa fa-trash-o" title="Delete"></i></span>

### JS:

function del\_properties(**e**){

**$(e)**.remove();

}

# Jquery UI

## Sortable:

### Option:

**appendTo**: ??????

**axis :** chỉ có thể di chuyển element theo chiều đã chỉ định (“x” : theo chiều ngang; “y”: theo chiều dọc)

**cancel:** không cho kéo để sắp xếp

**classes:** thêm class vào 1 class cụ thể là class của sortable

**connectWith:**  có thể sortable với những list sortable tương tự khác

**containment :** giới hạn vùng di chuyển, ko dùng chung với scroll. Giá trị là “id” của vùng thẻ html muốn giới hạn hoặc “parent” chỉ cha của đối tượng

**cursor**: quy định con trỏ chuột sẽ hiển thị như thế nào khi sortable

**cursorAt**: con trỏ chuột sẽ luôn xuất hiện tại vị trí (position) nào trên phần tử đang kéo khi sortable

**delay**: delay một khoảng thời gian trước khi được phép sắp xếp

**disabled**: disabled: true thì sẽ không thể disable được nữa

**distance**: phải kéo ra ngoài 1 khoảng bằng bao nhiêu đó thì mới thực hiện việc sortable được

**dropOnEmpty**: = false thì sẽ không thể thả element vào trong 1 empty connect sortable được

**forceHelperSize**: ???

**forcePlaceholderSize**: border giữ chỗ đúng bằng kích thước của item được kéo

**grid**: ???

**handle**: ???

**helper**: ???

**items**:

**opacity**: phần tử được kéo sẽ bị làm mờ đi. **opacity: 0.5**

**placeholder**: tạo khoảng không giữ chỗ cho item được kéo. **placeholder: "card-placeholder",** card-placeholder là một class. Hãy style lại nó. Dùng chung placeholder với forcePlaceholderSize để hiển thị đúng hơn.

**revert**: Làm cho việc thả một item vào vị trí của nó mượt mà hơn

**scroll**:

# Argular

* Không phải chữ số replace(/[^a-z0-9\s]/gi, '')
* Là khoảng trắng replace(/[\_\s]/g, '\_')

# UI – UX

## Lý thuyết

### UX tốt là như thế nào:

- Tạo được cảm giác về việc người dùng có thể điều khiển được (Ví dụ: iphone là một sản phẩm ai cũng có thể dùng được, từ người già đến trẻ con, từ đàn ông đến phụ nữ, và không mất quá nhiều công để sử dụng nó)

- Tạo được cảm giác an toàn (ok, tôi biết cái đó sẽ làm được, không vỡ hỏng điều gì cả và không làm ai bị thương hết, nó an toàn)

- Tạo cảm giác về tính tự ngã (Tính cá nhân của sản phẩm, ok tôi dùng nó vì không một ai có một thứ như thế, tôi dùng nó vì tôi thấy tôi mạnh mẽ hơn, sành điệu hơn, xinh đẹp hơn)

### Mô hình cầu thang

**1. Findable**

Người dùng có thể tìm thấy những gì họ cần tìm hay không? (Hệ thông tìm kiến, menu điều hướng, phân trang có dễ dùng hay không – Điều người dùng quan tâm đầu tiên là website này có bán sp mà họ cần hay không?)

**2. Accessible**

Người dùng có đủ điều kiện để dùng được sản phẩm hay không? (bạn có một website rất đẹp nhưng ko có phiên bản trên mobile thì bạn đã mất 60% người dùng sản phẩm. Hoặc website của bạn phải tải một video nền nên chạy quá chậm. Trung bình 10s mới tải xong thì khả năng người dùng rời bỏ bạn là rất cao)

**3. Usable**

Sản phẩm có dễ sử dụng không? (Suy cho cùng thì sẽ có rất nhiều cách để thực hiện một việc gì đó. Và người dùng sẽ chọn cách dễ dàng nhất và đơn giản nhất)

**4. Desirable**

Sản phẩm của bạn có lôi cuốn không và dễ nhớ không? (Đây là bậc thang mà giao diện người dùng sẽ thực hiện vai trò. Tại đây người thiết kế trải nghiệm sử dụng các hình ảnh, màu sắc hoặc các kỹ thuật sử dụng từ ngữ để thúc đẩy hành động hay là khơi dậy sự tò mò để khiến người dùng đi tiếp)

**5. Credible**

Sản phẩm của bạn có đáng tin cậy không? (1 thiết kế UX tốt phải xây dựng được 1 sự tin cậy, và đó cũng là một sự tin cậy quan trọng nhất của UX)

Biểu tượng màu xanh lá cho ta cảm giác an toàn hơn

**6. Useful**

Đây có phải là một sản phẩm có giá trị không? Nó có đáp ứng nhu cầu của tôi không? (Ví dụ: khi người dùng đặt hàng và trả tiền trên 1 website bán hàng của bạn, nhưng do hệ thống quản lý tồi, bạn buộc phải liên hệ và báo rằng “không có hàng ở trong kho” mặc dù trên website có niêm yết. Và bạn phải hoàn tiền lại cho khách hàng sau 7 ngày. Lúc đó khách hàng sẽ cho rằng, đây là một website mua hàng không uy tín và không có giá trị)

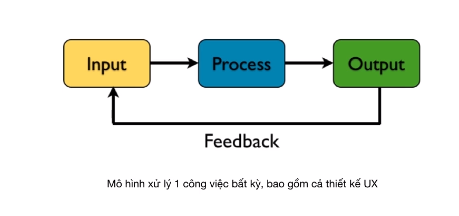
Có một điều đặc biệt khi vượt qua được bước cuối cùng này, người dùng đã thấy đây là một sản phẩm hữu dụng và đạt được user goal của họ thì hầu như bạn đã thành công. Thậm chí, đôi khi họ vướng mắc trong quá trình sử dụng nhưng thường là họ sẽ không bỏ bạn vì nó thực sự có ý nghĩa và cần thiết cho họ

Chúng ta phải đi lần lượt từng bước, từ thấp đến cao, khách hàng sẽ không nhảy 1 phát lên bậc thang useful được. Mỗi bậc thang phải cao vừa phải, không cao đến mức người phải bỏ cuộc.

Không nên thiết kế giao diện người dùng ngay từ lúc đầu mới vào

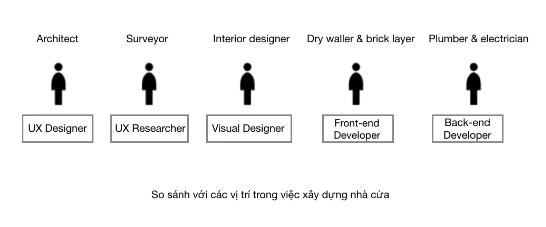
Các đơn vị làm phần mềm thường bỏ qua bước credible vì nó rất khó để thực hiện (có thể bạn sẽ phải có chứng chỉ về SSL, chứng chỉ về an toàn thông tin hoặc là xây dựng lâu dài thông qua dịch vụ khách hàng)

### 3 thứ tạo ra 1 ux tốt



Input: Người dùng cần gì và mục tiêu kinh doanh (Khả năng sai từ việc tìm ra input là thường xuyên xảy ra) Người dùng hay nói về những điều họ muốn chứ không nói được những điều họ cần. Nếu bạn không có 1 kỹ thuật khoa học để hiểu được những gì họ cần bạn sẽ rất dễ bị đánh lừa bởi những gì họ nói. Business need (tăng lượng doanh thu, tăng lượng khách hàng hay giảm tỷ lệ rời bỏ sản phẩm). User need and business need khó đi chung với nhau. Trong một số trường có thể làm tăng doanh thu nhưng làm giảm trải nghiệm người dùng. Công việc của UX chính là cân đối giữa hai yếu tố này

* Các vị trí (role) trong 1 team UX



**UX designer**: kiến trúc sư

Lãnh đạo team UX để đảm bảo mọi việc diễn ra đúng quy trình, làm việc với người dùng mục tiêu và các đơn vị liên quan đến sản phẩm (team marketing, phòng quảng cáo, phòng thư pháp, bộ phận bán hàng, bộ phận phát triển sản phẩm…). Mục tiêu là chuyển đổi yêu cầu của người dùng, yêu cầu kinh doanh thành đầu vào cho việc phát triển sản phẩm

Lo phần kiến trúc cho ứng dụng: thiết kế thông tin, tổ chức thông tin, xây dựng persona, tạo tags role hoặc user

**UX Researcher**: người giám sát công trình (cái gì khó thì hỏi người này)

Kiểm tra và nghiên cứu để hiểu tốt hơn về yêu cầu của người dùng và tìm ra các vấn đề. Nếu không có một người giám sát xây dựng thì ngôi nhà sẽ dễ bị sụp lún, nứt tường sau khi ở được vài năm.

**Visual designer**: thiết kế nội, ngoại thất (hình ảnh, màu sắc, phông chữ, biểu tượng, đáp ứng cả việc làm cho nó hoạt động tốt hơn, như xây một ngôi nhà, ngoài làm cho nó đẹp hơn thì còn làm cho nó tiện nghi hơn), làm cho sản phẩm dễ dùng và thông minh hơn

**Front-ent developer**: thợ xây gạch, trát tường (biến file thiết kế thành một sản phẩm hữu hình). Đảm bảo đầu ra với chất lượng pixel perfect đối với file thiết kế. Đảm bảo tính tương thích trên nhiều thiết bị. Đảm bảo tính bảo mật giữa các trang màn hình. Đảm bảo các hiệu ứng animation giữa các thành phần giao diện. Animation thực sự hấp dẫn với người dùng. Ux, UI không thể thay thế front-ent chính vì điều này

Backend developer: thợ hàn, thợ điện nước, đục bê tông, chạy đường điện, đường nước. Rất quan trọng trong team UX. Người dùng sẽ từ bỏ sử dụng website của bạn khi cần 10s để tải một trang chủ (cho dù trang đó có đẹp đến mức nào). Nhưng người vẫn ưu thích một website chạy nhanh và giao diện không cần bắt mắt

- 1 người như UI designer, UX designer là không đủ để sản phẩm được người dùng yêu thích

- Và có thể nói, một trong những nhiệm vụ lớn nhất của một designer giỏi chính là việc liên kết mọi người trong team thành một tập thể hỗ trợ lần nhau và tập trung vào trải nghiệm của người dùng. Những tính cách như làm việc cá nhân, chống đối sẽ hủy diệt một sản phẩm.

- Một UX designer có thể kiêm vị trí của UX researcher nhưng không nên kiêm vị trí của UI designer. Đối với 1 người làm UX tốt thì nên có kiến thức về Visual design nhưng thường thì không làm tốt ở vai trò đó. Nếu đặt người dùng vào yếu tố trung tâm thì cũng phải đặt UX designer vào vị trí trung tâm

### Vai trò của thiết kế đối với doanh nghiệp

Các nhận dạng như là hình ảnh, ngôn từ, tương tác, cảm xúc, sáng tạo trên sản phẩm của của riêng bạn người khác sẽ khó lòng bắt chước được => UX rất quan trọng, yếu tố để cạnh tranh giữa các doanh nghiệp

* Lợi ích của UX đối với doanh nghiệp

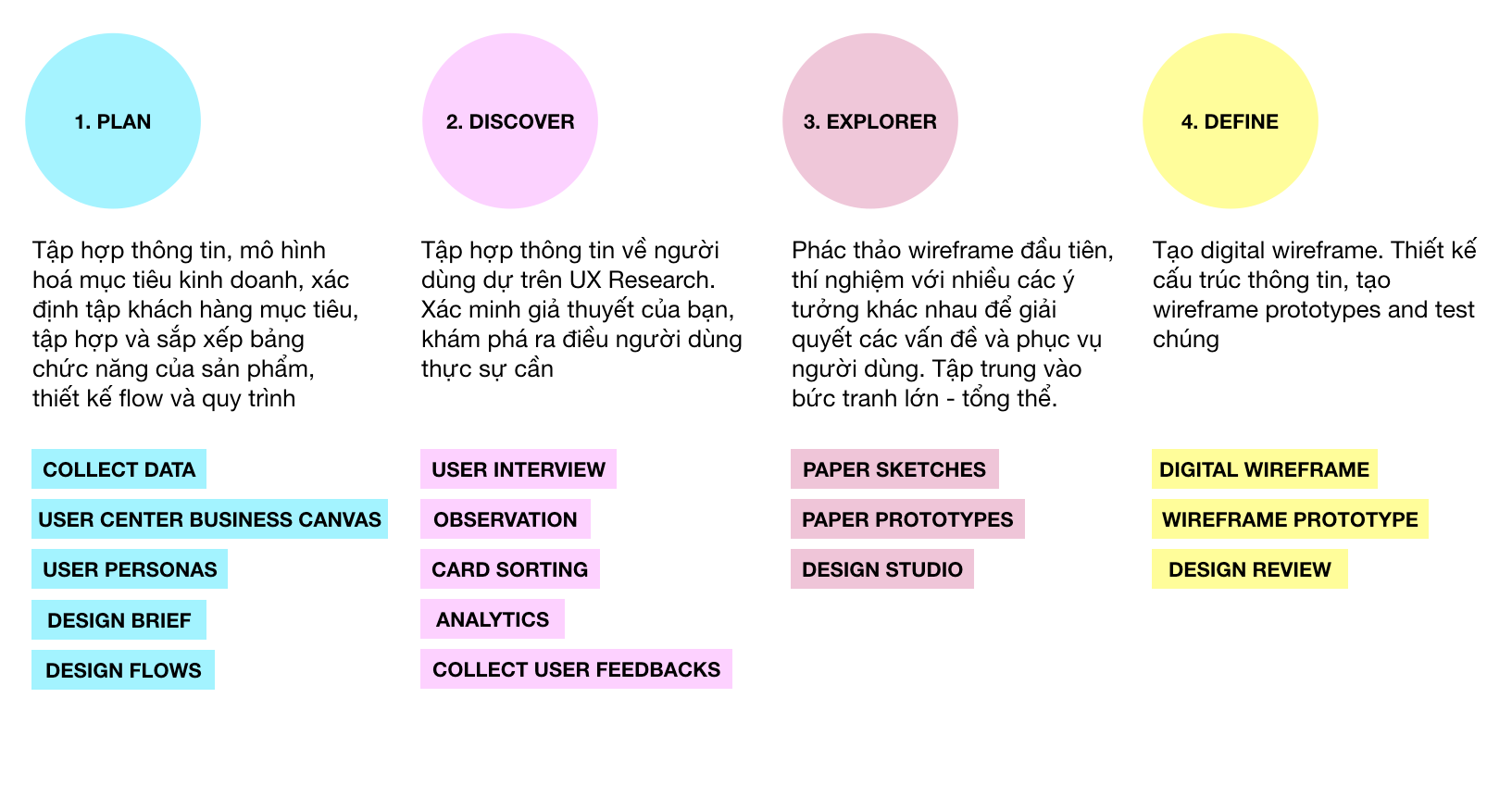
Giảm chi phí tạo ra sản phẩm: Có rất nhiều thông tin, cả những công ty start-up hay những công ty lớn gặp một vấn đề và họ phát triển một sản phẩm, họ hoàn toàn dựa trên các dự đoán, tưởng tượng. Họ nghĩ sản phẩm sẽ hoạt động như thê này, sp hoạt động như thế kia, và áp đặt rằng là người dùng sẽ cũng sẽ nghĩ giống họ. Và sản phẩm xây dựng gần xong thì mới đưa sản phẩm cho người dùng. Lúc đó, họ mới nhận thất một sai lầm. Và cái giá cho sai lầm đó đơn giản là thay đổi cảm nhận. Còn phức tạp hơn là đập đi làm lại từ đầu. Hãy đưa người dùng vào những đầu tiên để tránh sai lầm, lãng phí thời gian và công sức

* Giảm rủi ro khi nâng cấp, thay đổi sản phẩm
* Có cơ hội kiếm được nhiều hơn
* Sản phẩm trở nên dễ bán hơn

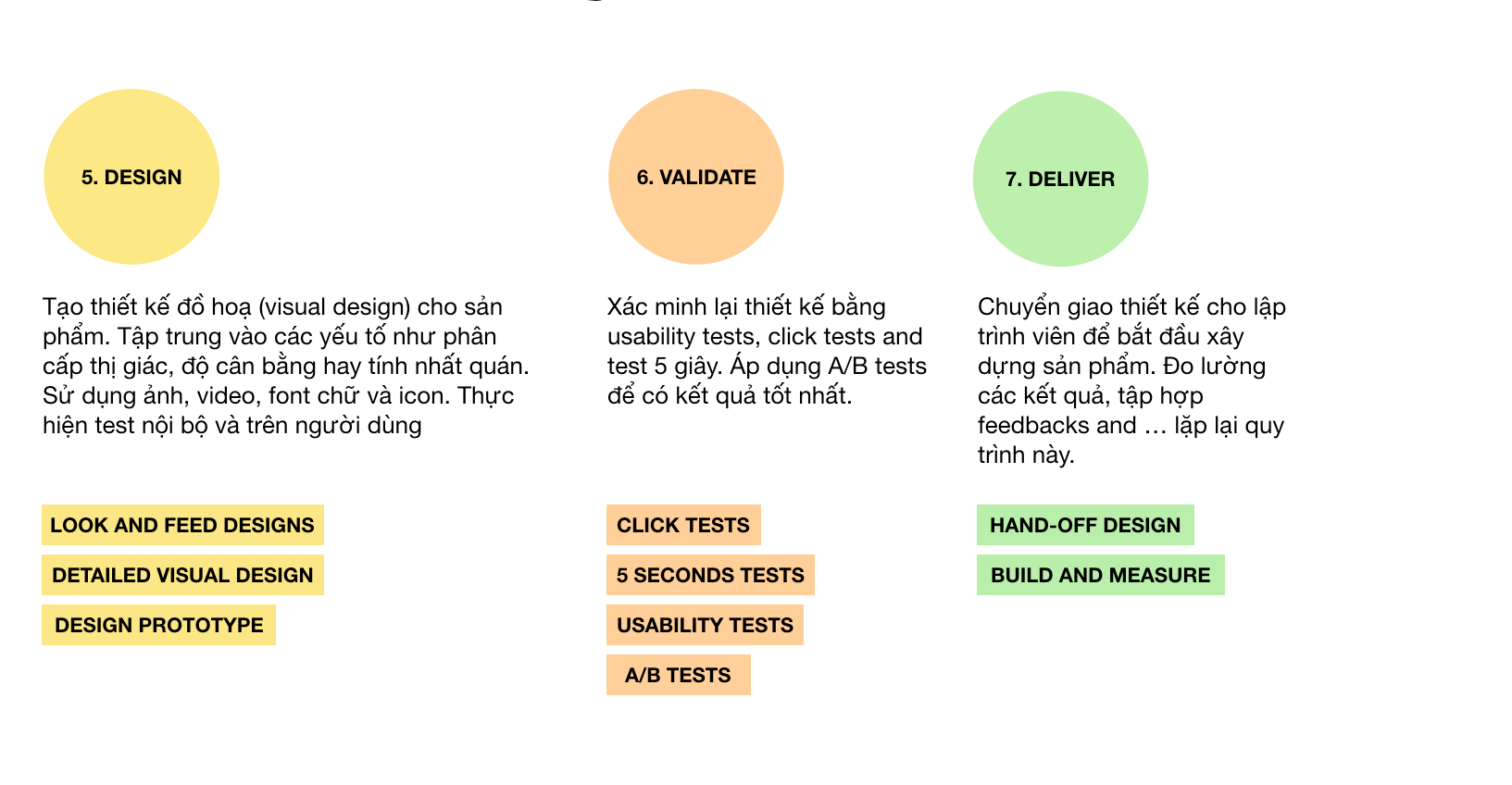
Trải nghiệm người dùng là tất cả về một quy trình chứ không có gì là cao siêu, khó hiểu hay là có một ý nghĩa nào khác

## Quy trình thiết kế UX

Bao gồm 7 bước:



Kỹ thuật và công cụ



* **Một số nguyên tắc**:

1. Lặp đi lặp lại

Chẳng hạn như khi vẽ xong wire frame ta phải thực hiện test với nội bộ và test với người dùng, sau đó ta lại sửa wire frame, tiếp tục lặp đi lặp lại quá trình đó, để tìm ra wireframe tốt nhất. Trong bước visual design cũng thế, ta cần phải thực hiện việc lặp đi lặp lại quá trình thiết kế rồi lại test, rồi thiết kế cho đến khi có được sản phẩm cuối cùng tối ưu



Ứng dụng một cách mềm dẻo

## Dự án mẫu

Giả thiết rằng chúng ta cần phải thiết kế một ứng dụng dating trên mobie (tức là 1 ứng dụng giúp kết bạn, giúp tìm hiểu nhau.. dành cho những người độc thân tại Việt Nam). Tên ứng dụng tạm gọi là vLover. Sản phẩm này được thực hiện theo yêu cầu của công ty VietLove, đã có thâm niên 5 năm trên thị trường môi giới hôn nhân. Công ty VietLove sau khi thành công ở thị trường offline, tức là đã môi giới thành công cho hàng nghìn cặp đôi, thì bắt đầu xây dựng app để tấn công trên thị trường online. Và đây là sản phẩm đầu tiên mà công ty này thực hiện để tiến vào thị trường. Và chúng ta sẽ cùng 1 team để thiết kế UX cho sản phẩm.

* Một số ý tưởng truyền tải:

- Đăng ký tài khoản miễn phí

- Tìm kiếm những người phù hợp

- Kết bạn và trò chuyện

* Ứng dụng sẽ kiếm tiền được bằng 2 cách:

- Thành viên VIP

- Dịch vụ môi giới hôn nhân

* Kỹ thuật

Ứng dụng sẽ chạy trên 2 nền tảng mobile là Android và iOS, nền web sẽ làm sau.

Chúng ta sẽ sử dụng những đầu vào này để thiết kế một sản phẩm UX tốt nhất.

### LẬP PLAN

Xác định phạm vi, dự án UX này là gì. Tạo ra một sp mới từ đầu, tối ưu UX cho một sp đã tồn tại, hoặc thêm một vài chức năng mới cho một sp. Project là một thuật ngữ ám chỉ một thứ cần phải áp dụng quy tình thiết kế UX để thực hiện

Một cách đơn giản, mục tiêu của bước lập PLAN sẽ bao gồm xác định 3 thứ:

User needs là gì (yêu cầu của người dùng)

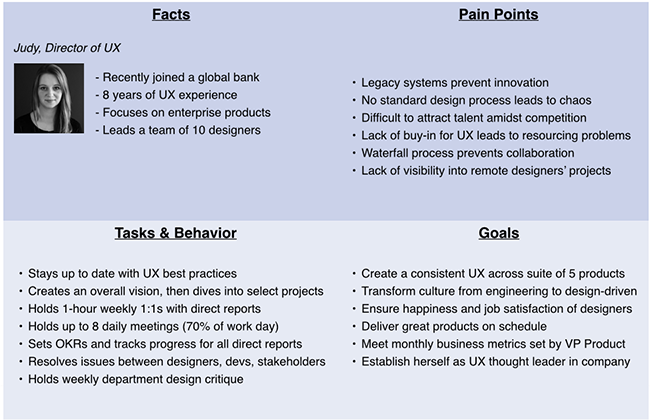
Mục tiêu kinh doanh (hay còn gọi là business needs là gì)

Những chức năng nào trên sản phẩm sẽ được triển khai thiết kế

Một trong những điều quan trọng nhất trong bước lập PLAN chính là việc tập hợp mọi người trong dự án lại với nhau, cùng nhau đóng góp ý kiến và cuối cùng là hiểu giống nhau về mục tiêu của dự án.

#### **Giai đoạn 1: Strategize**

Để có 1 cái nhìn sâu hơn và hiểu đúng mục tiêu của ngừoi dùng và những gì họ cần, chúng ta sẽ tạo ra User Personas. Và Personas cũng rất quan trọng để xác định người dùng mục tiêu và định thướng tìm người phù hợp để thực hiện test trong các bước tiếp theo. Và để đảm bảo giai đoạn này có thể đo lường được thì chúng ta sẽ xây dựng một ma trận mà áp dụng framework AARRR. Đây cũng là 1 phương pháp rất hay được nhiều công ty áp dụng trong thực tế và còn được gọi là Startup Matric và chúng ta sẽ đi sâu trong các bài sau.



*Hình trên: Tạo Personas cho dự án*

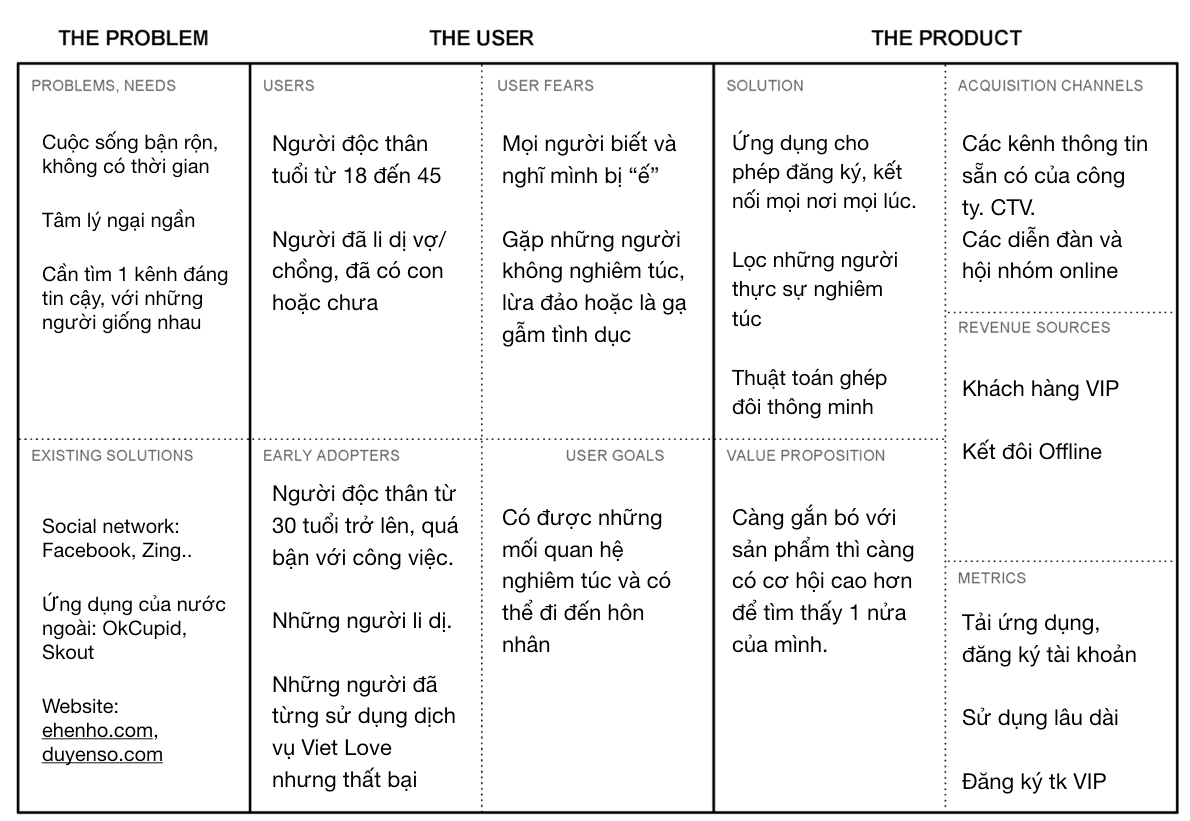
##### Mô hình User Centered Business Canvas (UCBC)

Lợi ích của mô hình này:

1 cái nhìn tổng quan

Dễ để hình dùng với tất cả mọi người

Tập trung vào những vấn đề cốt lõi



PROBLEM, NEEDS: Vấn đề của họ là gì? Họ sẽ cần gì ở sản phẩm của bạn

EXITING SOLUTIONS: Những giải pháp đã tồn tại

THE USER : Đâu sẽ là đối tượng khách chính. Những gì mà ta biết về họ

EARLY ADOPTERS: những khách hàng đầu tiên của chúng ta, họ là những người cao nhất có nhu cầu sản phẩm và sẵn sàng trả tiền

USER FEARS: Những điều mà khách hàng sỡ hãi và lo lắng khi dùng sản phẩm của chúng ta

USER GOAL: mục tiêu mà người dùng muốn sử dụng sản phẩm này

THE PRODUCT: Sản phẩm là gì? Giải pháp là gì để giúp người dùng xử lý vấn đề và đạt được mục tiêu

VALUE PROPOSITION: Điều gì khiến sản phẩm của chúng ta vượt trội và khác đối thủ

ACQUISITION CHANEL: Các kênh thông tin mà sẽ thu hút người mới

REVERNUE SOURCE: Những nguồn tạo ra doanh thu cho sản phẩm

METRIC: Làm thế nào để đo lường sự thành công của sản phẩm

##### Mô hình User persona

User persona là một nhân vật tưởng tượng tiêu biểu cho người dùng mục tiêu.

Một sp thông thường có nhiều persona

Tài liệu persona là một loại tài liệu thường xuyên phải được cập nhật

Trong dự án ví dụ thì có hai loại persona: 1. những ngừi độc thân, 2. những người đã từng có gia đình

Đặc điểm của user persona :

Là một bản tập hợp những thông tin quan trọng và hữu ích nhất về người dùng

Căn cứ vào user persona để tìm ra các ứng viên phù hợp, để thực hiện UX research

Thể hiện được yêu cầu và mục tiêu của người dùng

Căn cứ trên 1 người thật hoặc 1 nhóm người thật

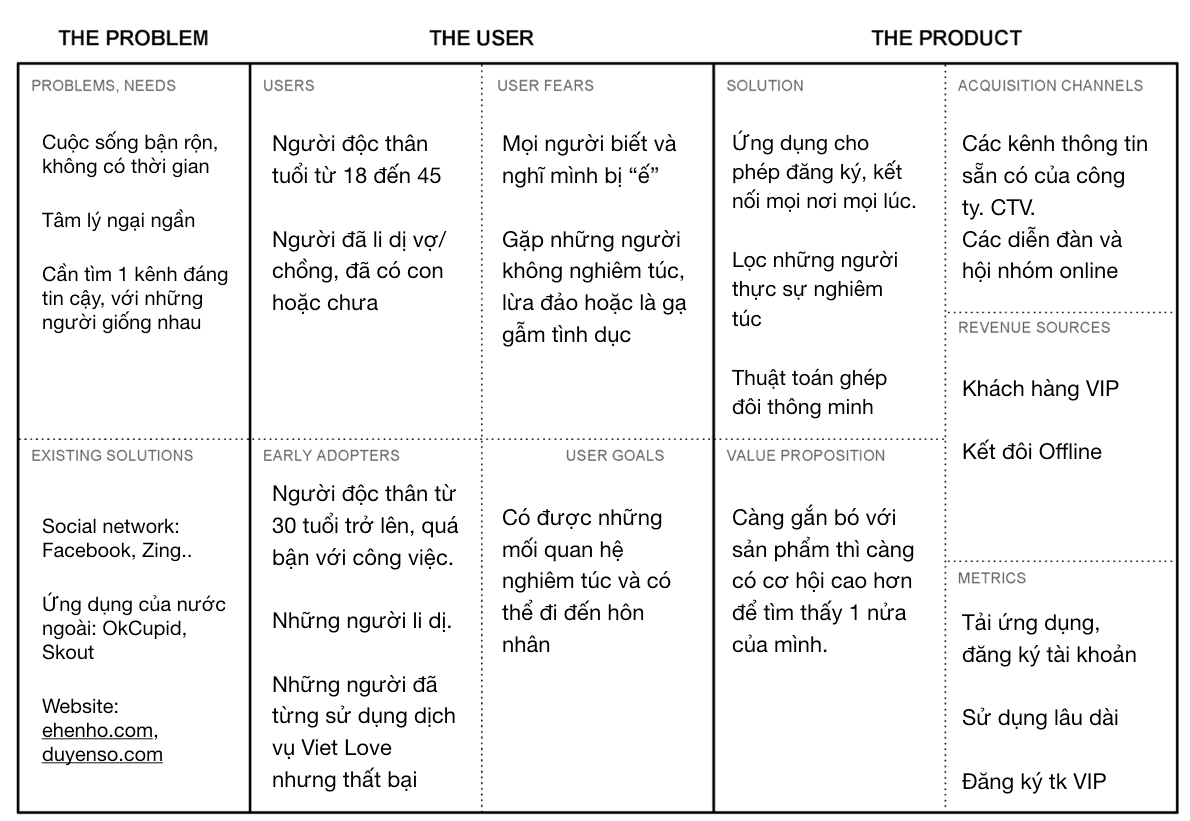
Vai trò của User persona

UX Designer định hướng thiết kế

UX Research tìm được ứng viên mục tiêu cho việc test, interview và research

Visual Designer xác định được phong cách thiết kế

Mẫu user personas đơn giản nhất



Sau khi có bộ user persona thì chúng ta cần chứng thực thông tin này là chính xác.

User Interview (phỏng vẫn người dùng)

Observation (Quan sát cách người dùng mục tiêu xử lý vấn đề)

Analytics, big data (Dựa trên những thông tin đã thống kê để xác thực thông tin)

Usability Tests (Trong quá trình ta test ứng dụng với khách hàng 🡺 là cơ hội xác thực thêm thông tin)

Customer Service (Phân tích trong các tài liệu hỗ trợ khách hàng, review)

UX Survey (Tạo các bài khảo sát và gửi đến tay người dùng mục tiêu)

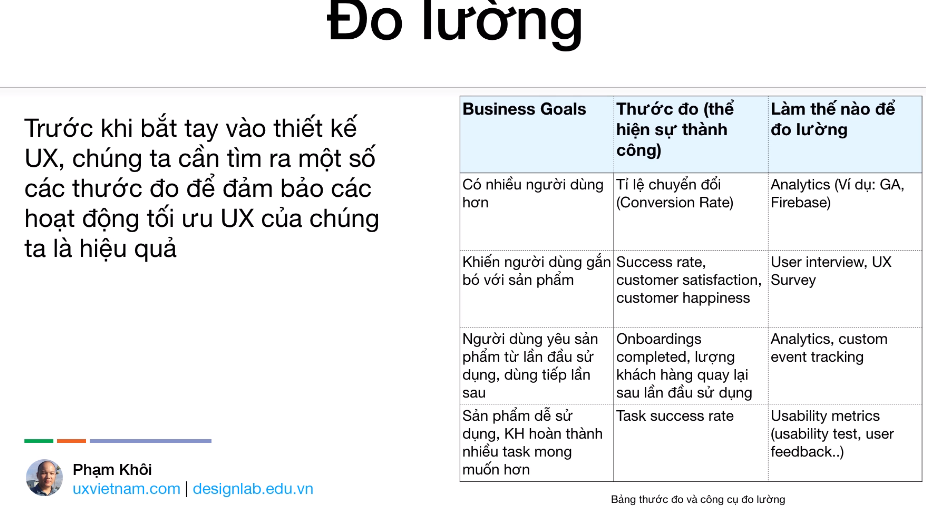
##### Mô hình AARRR

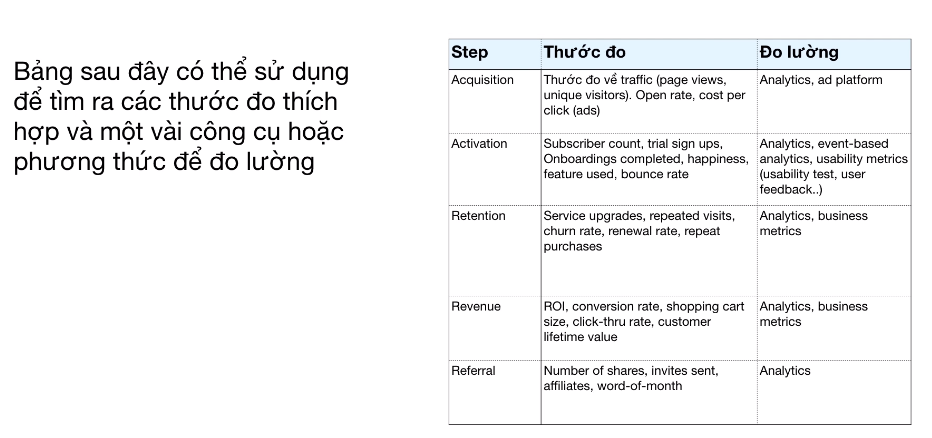
Đầu tiên, cần đảm bảo mục tiêu kinh doanh là gì. Và đảm bảo toàn bộ team của chúng ta đồng ý.

Đảm bảo mục tiêu kinh doanh thì rõ ràng và khả thi (tăng convertion rate, tăng returning visitor)

Mô hình AARRR bao gồm:

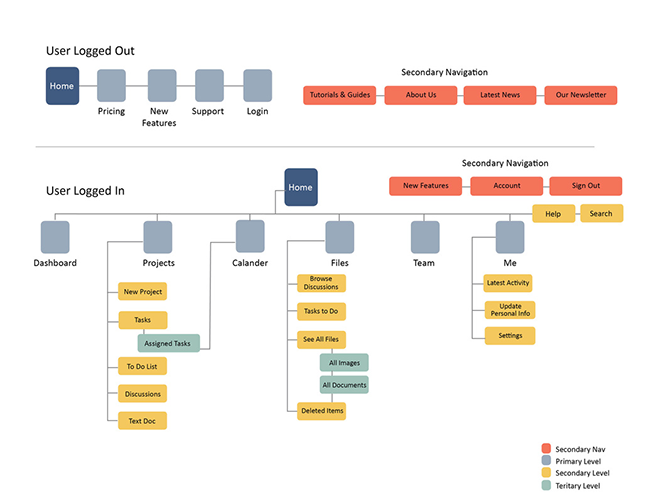
* Acquistion (làm cách nào mà người dùng có thể tìm đến sp của bạn? SEO? Email compaigns? )
* Activation (người dùng bắt đầu sử dụng sản phẩm. Ví dụ: tải app từ store, đăng ký thành viên hoặc mua 1 sp nào đó. Trải nghiệm đầu tiên đối với sp ra sao. Họ có thực sự bắt đầu sp ko?. họ có đáp ứng được các mục tiêu kinh doanh hay không? – Giải pháp: hỗ trợ thêm màn hình hỗ trợ, tài liệu hướng dẫn để người dùng dễ dàng sd. Hoặc đơn giản hóa việc đăng ký, và cho họ bổ sung thông tin sau)
* Retention (Sau khi người dùng sd sp của bạn thì họ sẽ quay lại hay ra đi mãi mãi. Chúng ta có thể gửi email cho khách hàng hoặc )
* Revenue (Đề cập đến việc kiếm nhiều tiền nhất từ sản phẩm). Tăng tỷ lệ chuyển đổi. Tăng cường các hoạt động giảm giá, tặng voucher. Thường xuyên đề xuất và điều hướng người dùng đến những sản phẩm mục tiêu
* Referral (Người dùng có nói về sản phẩm của bạn với người khác, như là bạn bè hay gia đình hay không). Ví dụ như các công cụ chia sẻ trên mạng xã hội, gửi link qua email, hay tổ chức các chương trình maketing tặng thưởng



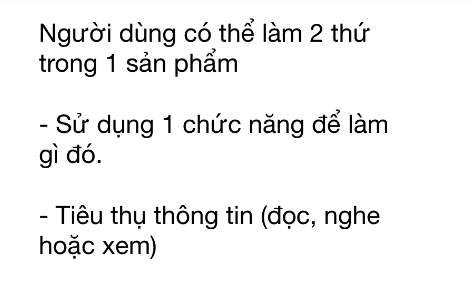


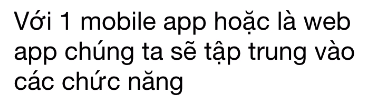
#### **Giai đoạn 2: Organize**

Bây giờ là lúc chúng ta cần phải ngồi lại để sắp xếp lại chúng, quyết định xem cái gì nên được design và làm cách nào để design. Giai đoạn này chúng ta sẽ thiết kế về functionality map, sitemaps và Design Brief.



*Hình trên: Sitemap là một công cụ hữu hiệu để tổ chức thông tin*



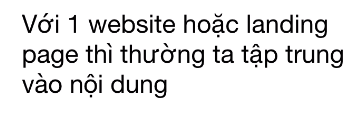


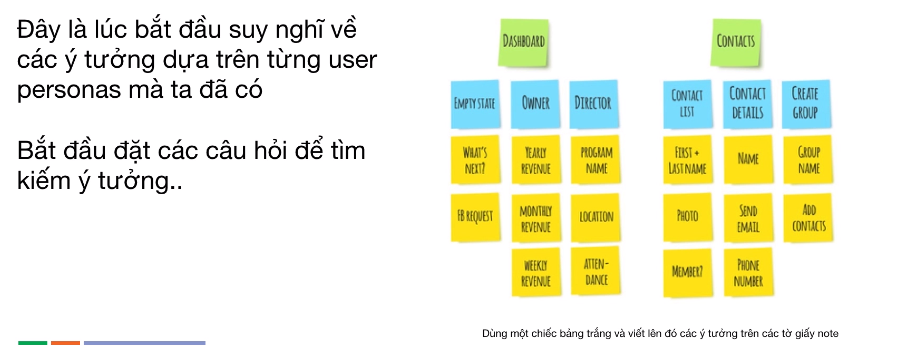
Ví dụ về ứng dụng Vlover:

- Đăng ký

- Tìm kiếm người phù hợp

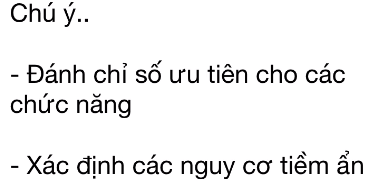
- Giao tiếp với nhau

🡺 Nội dung ở đây là hình ảnh, văn bản, video



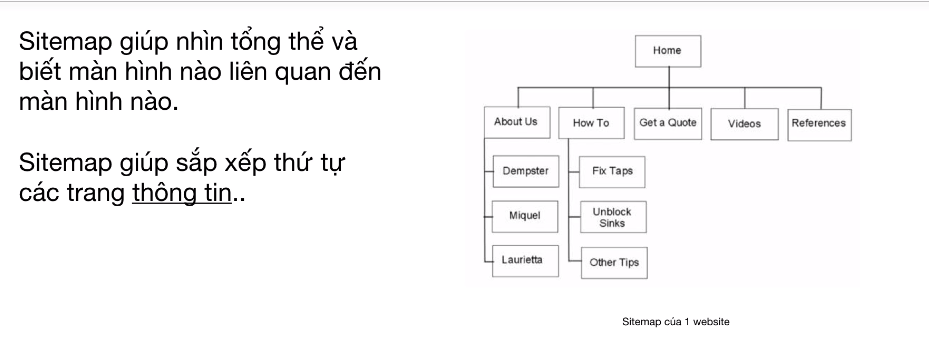
Những câu hỏi như:

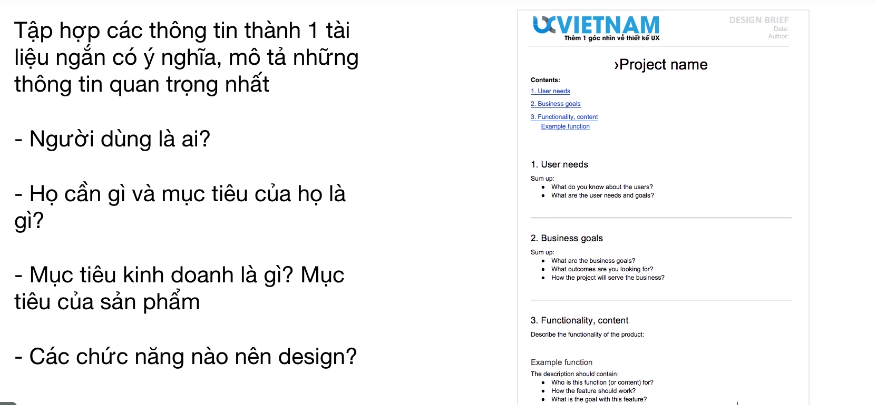
* Chức năng nào sẽ giải quyết được vấn đề của người dùng này
* Chức năng nào mà người dùng đang tìm kiếm
* Nội dung nào mà người dùng này cần (văn bản hay video)
* Nội dung nào giúp người dùng không còn sợ hãi hay lo lắng
* Thông tin nào mà người dùng đang tìm kiếm



* Có thể có các chức năng có vấn đề với thông tin cá nhân nhạy cảm của người dùng
* Các chức năng quá tốn công sức mà chưa chắc người dùng đã sử dụng



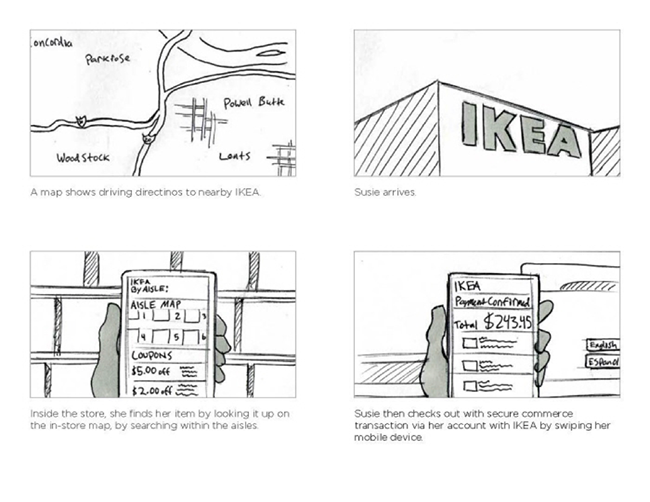




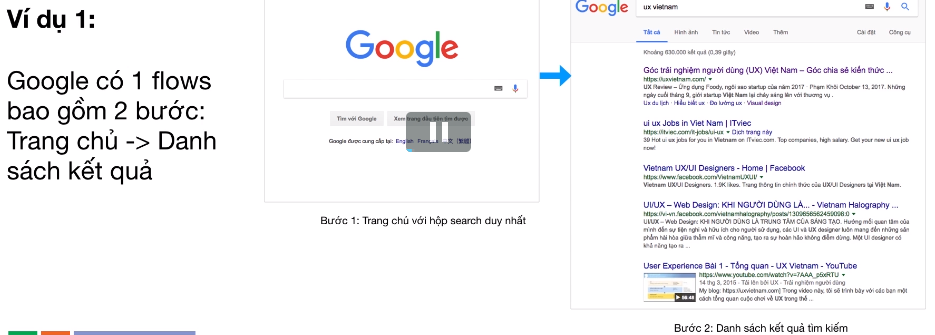
Để có được thông tin này thì ta có thể trích xuất từ bảng business goals mà ta đã học từ bài trước

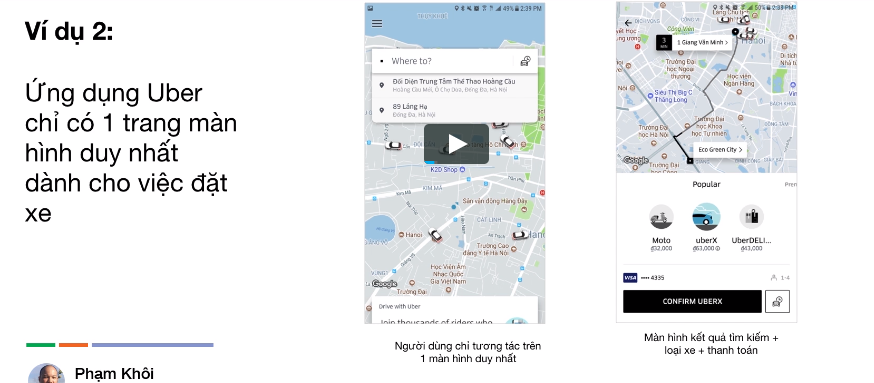
#### **Giai đoạn 3: Design Flows**

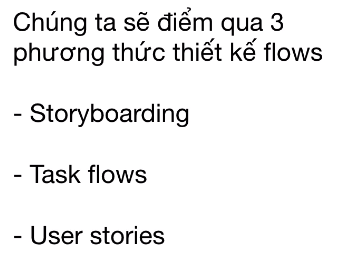
Chúng ta sẽ tập trung vào xây dựng các flow cho dự án. Flow (hay còn gọi là luồng) là con đường từ ngữ cảnh này chuyển sang 1 ngữ cảnh khác. Một trong những trải nghiệm người dùng quan trọng nhất của 1 sản phẩm phần mềm chính là flow. Vì sao lại vậy? bởi vì dù bạn có 1 thiết kế giao diện đẹp và ấn tượng đến đâu, mà flow không tối ưu thì cũng trở nên vô nghĩa. Trong giai đoạn này chúng ta sẽ tập trung vào việc xây dựng các flow của sản phẩm, sẽ có những phương thức như là: user stories, storyboarding and task flows



*Hình trên: Design Flows - Mô hình hóa flow của người dùng bằng hoạt hoạ*







**Storyboarding**:

Một phương thức thú vị và giản đơn được mượn tượng từ công nghiệp làm phim đề chúng ta đưa một câu chuyện vào cuộc sống. Mục tiêu củ storyboarding là tập trung vào ngữ cảnh, tức là ở đâu, khi nào, người dùng sử dụng sản phẩm như thế nào. Điều quan trọng là story boarding chỉ tập trung vào những gì người dùng cần và mục tiêu của họ thay vì nghĩ đến các giải pháp kỹ thuật cụ thể.

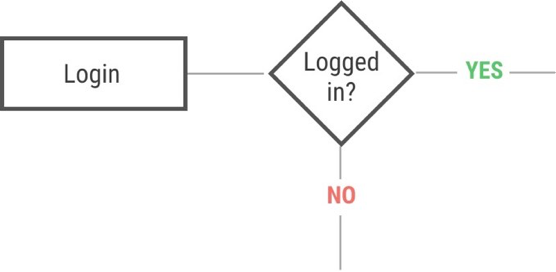
Những ai đó có thể bảo: “tôi không biết vẽ”. Thực tế đây chỉ là một loại hình vẽ đơn giản vui nhộn. Ta sẽ chia thành nhiều ô, mỗi ô là một bước trong flow. Trong mỗi ô ta sẽ thể hiện ngữ cảnh, tư duy và nhu cầu của người dùng, sau đó kèm theo một đoạn test để giới thiệu. Với mọi chức năng. ta chỉ nên vẽ toàn bộ các bước trê một mặt của tờ giấy A4 là đủ. Nếu như bạn cần đến một tờ A3 để vẽ 1 storyboarding thì tức là flow của chúng ta đã trở nên quá phức tạp và cần phải xem xét lại.

Tại sao chúng ta lại cần story boarding:

- Thực tế thì thời điểm chúng ta vẽ nghệch ngoạc ra giấy, chính là lúc ta tư duy tốt nhất. Khoa học đã chứng minh đó là một cách để sáng tạo, tìm kiếm các ý tưởng, rất fun, rất relax. Vẽ tay là một việc mà mọi người trong team đều có thể làm được

**Taskflow:**

Nếu như Storyboarding chỉ tập trung vào ngữ cảnh và người dùng, không hề đếm xỉa đến sản phẩm và yếu tố kỹ thuật thì Task Flows lại tập trung vào điều đó. Đây là mô hình sẽ thể hiện những việc mà người dùng cần thực hiện trên sản phẩm để hoàn thành một tác vụ.



*Hình trên: Task flows*

Để xây dựng 1 cái task flows chúng ta chỉ sử dụng 3 loại hình vẽ: chữ nhật: tên của trang màn hình hoặc là tác vụ của khách hàng (ví dụ như: đăng ký, đăng nhập hay tìm lại mật khẩu…). Hình thoi thể hiện điều kiện rẽ nhánh, thông thường có 2 trạng thái là YES, NO. (Ví dụ: Nếu như đã đăng nhập hay là nếu nhập sai mật khẩu 3 lần..). Hình thứ 3 là các mũi tên liên kết giữa các hình chữ nhật và hình thoi trong biểu đồ thể hiện luồng màn hình.

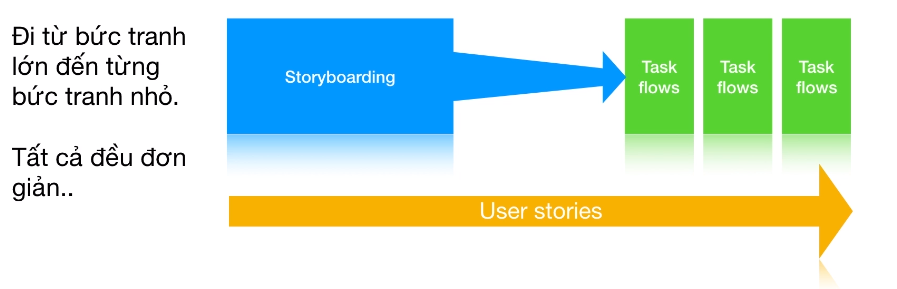
Sử dụng phần mềm để vẽ: visio, draw.io

**User stories**:

Đây là một mô hình nổi tiếng nếu như bạn biết đến phương pháp xây dựng phần mềm Agile. Phương thức này là 1 tập hợp các câu đơn, theo cấu trúc giống nhau: Là [ai đó..], tôi muốn [làm điều gì đó..], do đó [tôi cần..]. Ví dụ như: Là một dùng trên vLover, tôi muốn tìm được một vài người phù hợp để nói chuyện, do đó tôi cần 1 công cụ tìm kiếm tốt. Hay là: là một quản trị viên của vLover, tôi muốn kiểm soát chất lượng thông tin trên diễn đàn của vLover, do đó tôi muốn có 1 công cụ để người dùng cảnh báo hành vi vi phạm và gửi đến tôi.

Một user stories có thể ở mức độ rất high-level như trong ví dụ trước đó, hoặc ở mức độ rất cụ thể và chi tiết khi ta đi sâu phân tích từng chức năng. Nhưng nó đều tập trung toàn bộ vào user needs, luôn luôn đặt người dùng vào giữa trong mọi hoàn cảnh.

Cũng giống như Storyboarding và Taskflow, cấu trúc câu của User stories vô cùng giản đơn nhưng lại rất nhiều sức mạnh. bạn có thể tổ chức các workshop để tập hợp các user stories từ mọi người và xây dựng bổ user stories cho dự án.



## Link

Forum thảo luận: forum.uxvietnam.com

# Javascript

## Các hàm thường dùng

Cắt chuỗi: **str.split();**

Ép kiểu Int: **parseInt(); 🡺 parseInt(price)**

Ép kiểu String: **x.toString();**

# FloatThead

**headerCellSelector**: chọn ra cột header cụ thể

floatTableClass: