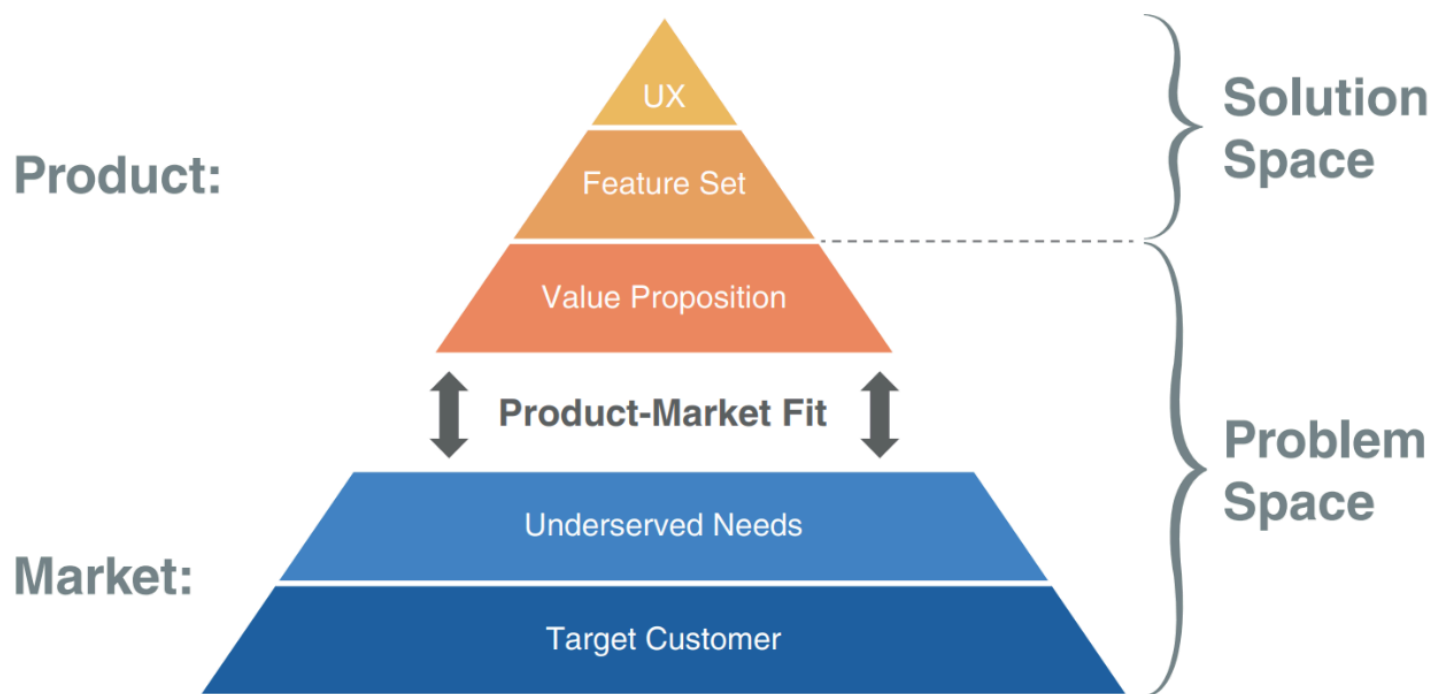




	NỘI DUNG	LƯU Ý
	Buổi 2: SDLC & PMF Overview <ul style="list-style-type: none"> Software development life cycles: 	

	<ul style="list-style-type: none"> o Analysis: phân tích xem mình sẽ làm những gì o Design: mình sẽ thiết kế phần mềm như thế nào o Develop: mình sẽ code ra sao o Test: phần mềm được đem đi test o Deploy: phần mềm được đưa vào sử dụng o Maintain: giai đoạn bảo trì, hỗ trợ khách hàng sử dụng phần mềm. <ul style="list-style-type: none"> ● Giới thiệu về PMF, gồm 5 layers chính: <ul style="list-style-type: none"> o Target Customer (Nhóm khách hàng có đặc điểm chung tương đồng nhau tiềm năng) o Underserved Needs (Những nhu cầu mà thị trường chưa phục vụ tốt) o Value Proposition (Giá trị mà sản phẩm sẽ giải quyết cho khách hàng là gì) o Feature Set (Các tính năng mà sản phẩm đấy đem lại value cho khách) o User Experience (Trải nghiệm giúp tối ưu hóa hành trình sử dụng khách hàng) 	
Target Customer	Buổi 3 _ Target Customer (Potential Segmentation) <ul style="list-style-type: none"> ● Demographic ● Behavior ● Need ● Psychology ● Attitude 	Buổi 3 còn học thêm về Product Life Cycles cơ bản: <ul style="list-style-type: none"> ● Introduction ● Growth ● Maturity ● Decline
Underserved Needs	Buổi 4. Đo lường phỏng vấn định lượng. <ul style="list-style-type: none"> ● Customer Value Delivered = Importance × Satisfaction ● Opportunity to Add Value = Importance × (1 – Satisfaction) ● Customer Value Created = Importance × (SAT_{after} - SAT_{before}) 	Lưu ý về số lượng người phỏng vấn: <ul style="list-style-type: none"> ● Phỏng vấn định lượng: theo các nghiên cứu khóa học thì đạt được độ

	<p>Buổi 5. Phỏng vấn định tính</p> <ul style="list-style-type: none"> ● Ice-breaking building rapport: quy trình thực hiện phỏng vấn ● Output: <ul style="list-style-type: none"> ● Empathy map ● User Persona <p>Khi đặt câu hỏi tránh đặt các câu hỏi đóng như Yes / No. Tận dụng các câu hỏi mở và đi sâu vào một nhánh insight nào đấy mà mình thấy có ích. Apply 5 Whys để đào sâu tìm Root Cause. Đối với 5 Whys khi sử dụng hỏi về Pain points thì final tìm được ra là Negative Emotion.</p> <p>Khi hỏi về các ký ức của user thì trigger não phải thay vì não trái bằng các câu hỏi: <i>“Hãy nhớ lại thời điểm gần đây nhất mà bạn...”</i>. Hoặc <i>“Lần gần cuối cùng mà bạn...”</i>. Không nên hỏi dồn dập và tạo tâm lý áp đảo user. Hãy khiến body của user trở nên thoải mái, quan sát cử chỉ tay, vai và ánh mắt của user.</p>	<p>uy tín cơ bản thì Sample Size > 30 người - observations (tốt nhất trên 125 người), tuy nhiên người làm innovation cho rằng nên từ 180 đến 3 ngàn người -> WHAT, HOW</p> <ul style="list-style-type: none"> ● Phỏng vấn định tính: trên 5 người (theo Nielsen thì phỏng vấn định tính từ 5 người sẽ cover được 85% Insight. Do có tính chất Diminishing Return -> WHY ● Nếu làm trên nhiều segment thì phải đảm bảo mẫu số giống nhau ở từng segment.
Value Proposition	<p>Buổi 6. Value Proposition Canvas</p> <p>Các bước để xây dựng Value Proposition Canvas như sau:</p> <p>Xây dựng Customer Profile trước.</p> <ul style="list-style-type: none"> ● Phỏng vấn tìm Customer Jobs (Job to be done đề xuất bởi Ulwick) ● Xác định xem có pain points gì ở Job Step đấy không ● Xác định Gain mà user mong muốn đạt được khi giải quyết được pain point đấy. <p>Xây dựng Value Proposition</p> <ul style="list-style-type: none"> ● Customer Jobs -> Products & Services ● Pains -> Pain Relievers 	<ul style="list-style-type: none"> ● Customer Jobs = Job to be done : V + Object -> đây là các hành động để đạt được Gain. ● Gain: Mong muốn gì ● Pain Point: trái ngược với Gain -> thường hỏi Pain Points để biết được Gain

	<ul style="list-style-type: none"> Gains -> Gain Creators 	
Feature Set	<p>Buổi 8. Feature Set & Backlog</p> <ul style="list-style-type: none"> Feature Sets: Danh sách các tính năng của sản phẩm theo độ ưu tiên: Must have, Performance, Delighter (<i>các tính năng Must Have phải có để build MVP</i>). User Story: Breakdown các Features ra nhiều User Stories. Ví dụ: <ul style="list-style-type: none"> Feature_Tạo đơn hàng User stories: Tạo đơn hàng ở trạng thái draft cho phép chỉnh sửa, Tạo đơn hàng ở trạng thái submitted không cho phép chỉnh sửa, Chỉnh sửa đơn hàng ở trạng thái Draft.... Tip define tính năng must have: CRUD_ Create, Read, Update, Delete Product Backlog: Bao gồm tất cả User Stories và các task khác liên quan của DEV, TEST để hoàn thành User Stories đó. Sprint Backlog: Theo Scrum Framework, mỗi sprint có timebox từ 1 đến 4 tuần -> gather các User Stories vào từng sprint để chạy tương ứng. Ví dụ: <ul style="list-style-type: none"> Sprint 1: Tính năng về Tạo, Sửa, Xóa Sprint 2: Các tính về Phê duyệt, Từ chối (Phải Tạo đơn trước -> Duyệt/Từ chối được). Scrum Framework: <ul style="list-style-type: none"> 1 Scrum team gồm: DEV, TEST, PM_<i>Project Manager</i>/Scrum Master/PO, ITBA. (Có nơi gọi là Development Team, Squad Team) 4 events chính thống trong Scrum Guide: 	<p>Theo mô hình PMF: Define Feature Set trước -> UI/UX sau.</p> <p>Mô hình này sẽ áp dụng tùy doanh nghiệp thôi: PO define Feature set trước -> ITBA cover flow, màn hình sau.</p> <p>Thực tế các dự án chỉ làm: Khi làm HLR phải define luôn có bao nhiêu màn hình, các tính năng trên màn hình tương ứng để ra được WBS báo giá khách hàng. Ví dụ về WBS (tài liệu dự án, không share ra bên ngoài nhé):</p> <div data-bbox="1596 868 1669 933" data-label="Image"> </div> <p>[Project Name]_Requirement '</p> <ul style="list-style-type: none"> WBS có 2 sheet về Functions và Screens. <p>Product Backlog gồm:</p> <ul style="list-style-type: none"> Tasks về màn hình: Screen Description. Tasks về chức năng: Use case Specific/User Story

	<ul style="list-style-type: none"> ▪ Sprint Planning / Sprint Grooming: Dựa vào Product Backlog, collect các User Stories vào Sprint Backlog. ▪ Daily Scrum: Họp đầu ngày check lại task công việc của từng thành viên. Timebox thường diễn ra 15 phút, và thành viên phải đứng cho tỉnh táo, do họp vào buổi sáng. Mỗi thành viên sẽ trả lời cho 3 câu hỏi: <ul style="list-style-type: none"> • Hôm qua làm gì • Hôm nay làm gì • Có gặp khó khăn gì chưa được xử lý không? ▪ Sprint Review / Sprint Demo: Demo sản phẩm cuối sprint cho Stakeholder. ▪ Sprint Retrospective: Họp review nội bộ trong team để rút kinh nghiệm cho sprint sau. 	<p>Khi phát triển: phải code màn hình trước -> có màn hình rồi thì mới code tiếp các tính năng trên màn hình đó.</p> <p>Ngoài ra Product Backlog có thể bao gồm:</p> <ul style="list-style-type: none"> - Tasks của phòng ban khác collab vs product - Support client - Maintenance - Fix Bug - Issue - ...
UX	<p>Buổi 7. Design Solution_Sketching</p> <ul style="list-style-type: none"> • Sketch -> Wireframe -> Mockup -> Prototype (https://thinhnotes.com/chuyen-nghe-ba/phan-biet-sketch-wireframe-mockup-va-prototype/) • Wireframe có Low-Fidelity Wireframe (thường test trong internal) và define rules UX với team UX Research (nếu có) • High-Fidelity Wireframe dùng để check internal và external được. Test bố cục khi phối màu và chữ để User có thể hiểu. Nên define tiếp tục với UX Team • Sitemap: define danh sách các màn hình và flow truy cập vào từng hình -> menu thứ tự các màn hình. <p>Buổi 9. User Journey_User Flow_ Wireflow</p>	<p>Workflow_Flowchart_BPMN</p> <ul style="list-style-type: none"> • Workflow tương đương với Process, thể hiện 1 quy trình làm việc, hay 1 quy trình nghiệp vụ nào đó. Để thể hiện được workflow sẽ dùng Diagram: <ul style="list-style-type: none"> ◦ Flowchart ◦ BPMN • Cả 2 loại Flowchart và BPMN đều đều thể hiện các bước thực hiện trên hệ thống và ngoài hệ thống, tương ứng với nghiệp vụ thực tế. • Sự khác nhau đó là BPMN sử dụng các ký hiệu theo 1 chuẩn ISO, người có hiểu biết về

	<ul style="list-style-type: none"> • User Journey: mô tả trải nghiệm cấp cao, toàn diện của người dùng trên các kênh (bao gồm cả tương tác với phần mềm & tương tác với bên ngoài) và theo thời gian. • User Flow: phóng to để mô tả một tập hợp các tương tác cụ thể, riêng biệt tạo nên lộ trình chung của người dùng thông qua một sản phẩm. • Wireflow: Wireflows là một định dạng đặc tả thiết kế (design-specification) kết hợp các thiết kế bố cục trang kiểu wireframe với cách thể hiện các tương tác giống như Flowchart. <p>Vậy Flowchart là gì? -> xem thêm ở cột “LƯU Ý” để hiểu thêm</p>	<p>BPMN đọc sơ đồ sẽ hiểu được ký hiệu nghĩa là gì. Còn Flowchart sử dụng các ký hiệu không theo chuẩn, người vẽ cần chú thích thêm về ý nghĩa của từng ký hiệu để người đọc có thể hiểu được sơ đồ.</p> <p>-> Phân biệt Workflow, User Flow:</p> <ul style="list-style-type: none"> • Nếu như Workflow nói đến 1 quy trình tổng quan có sự tương tác của nhiều người dùng (nhiều role). • Thì User flow nói lên flow của một người dùng (1 role cụ thể) tương tác với các màn hình trên ứng dụng phần mềm mà họ sử dụng (theo thứ tự của 1 prototype). • Từ workflow sẽ define ra nhiều User flow tương ứng. <p>Các loại Flow và nghiệp vụ ITBA sẽ được học sâu ở Level 2</p> <p>TÀI LIỆU_Output của ITBA/PO:</p> <ul style="list-style-type: none"> • HLR_High level Requirement (với công ty Outsource như FPT, CMC..., sau quá trình khảo sát với khách hàng, output của ITBA dùng để chốt với PO (thuộc bên KH) là HLR. Với công ty Product, PO sẽ chốt HLR với Stakeholder chịu trách nhiệm chính với sản phẩm này, vd: CEO..) ○ ORD: object relationship Diagram ○ Workflow (thể hiện qua diagram: flowchart hoặc BPMN)
--	---	--

		<ul style="list-style-type: none"> ○ Màn hình (UI) ○ Sitemap ○ Functions List (<i>breakdown nhỏ ra của Feature set</i>) -> Use case Diagram ○ Permission Matrix (ma trận phân quyền cho từng nhóm người dùng) ● Tài liệu chi tiết transfer cho DEV, Tester: SRS <ul style="list-style-type: none"> ○ Mô tả màn hình -> Screen Description ○ Mô tả chức năng -> Use case Specific
PMF	Buổi 11. Build MVP Test <ul style="list-style-type: none"> ● Flow: PoC -> Prototype -> MVP -> Production ● Phân biệt POC và Prototype <ul style="list-style-type: none"> ○ POC: Kiểm tra 1 ý tưởng, 1 tính năng của sản phẩm có thể phát triển khả thi có tốn quá nhiều thời gian hay không?-> PO/ITBA trao đổi với DEV ra được POC ○ Prototype -> PO/ITBA làm việc với Designer ra được Mockup, Prototype. <ul style="list-style-type: none"> ▪ Thay vì những màn hình tĩnh như mockup, với Prototype, người dùng hoàn toàn có thể tương tác trực tiếp với nó. ▪ Giúp khách hàng mường tượng rõ hơn về sản phẩm họ sẽ nhận được. 	Product Designer sẽ define: User Flow, Task Flow, User Action kỹ để dựng Prototype cho chuẩn.

Khi nói chuyện với BOD hoặc các Stakeholder thì họ sẽ đề cập đến MVP và các cách kiểm tiền MVP ra sao -> thì hiểu rằng đang đề cập đến MVP sau khi improve liên tục từ các MVP Tests hoặc Pretotype. Nhưng các bạn cứ hiểu Pretotype ~ MVP Prototype ~ MVP Test để tránh confuse nhiều thuật ngữ.

MVP Tests như Dan Olsen đề cập thì sẽ có methods nhưng để dễ nhớ thì hãy nhìn vào MVP Matrix xác định xem Goals (Product / Marketing) và Research triển khai (Qualitative / Quantitative) của mình là gì để chọn một phương pháp phù hợp. Kinh nghiệm của mình thì nên làm các MVP Test bằng Product các bản vẽ trước để clear về mặt Layout và UI trước khi code. Việc này giảm thiểu rủi ro các Dev phải fix bug liên tục. Song song quá trình test MVP Tests Product thì nên test bằng các Marketing MVP Test để xem mức độ quan tâm của user. Ví dụ khi improve xong 3 mockup nào đấy các. Các bạn có thể kết hợp Fake Door vs Landing Page để xem user có quan tâm và feedback ra sao (có một vài tool giúp user thu thập feedback tại landing page)

Buổi 10. Usability Testing

- **Định nghĩa:** Là một loại nghiên cứu người dùng nhằm đánh giá trải nghiệm của người dùng khi tương tác với trang web hoặc ứng dụng. Nó giúp các nhà thiết kế và nhóm sản phẩm của bạn đánh giá mức độ trực quan và dễ sử dụng của sản phẩm.
- **Mục đích:** tạo ra một sản phẩm giải quyết các vấn đề của người dùng và giúp họ đạt được mục tiêu với trải nghiệm tích cực.
- **Quy trình thực hiện Usability Testing:** Dựa vào **user Journey thực tế tương tác của người dùng với sản phẩm** (SP ở dạng Prototype hoặc MVP hoặc thậm chí Production) -> **Quan sát “Hành động”**,

“Cảm xúc” của người dùng -> Đặt “Câu hỏi” ở từng bước để ra được “Pain Point” -> Từ đó ra được giải pháp khắc phục.

Để tạo ra Usability Testing thì cần các thông tin sau:

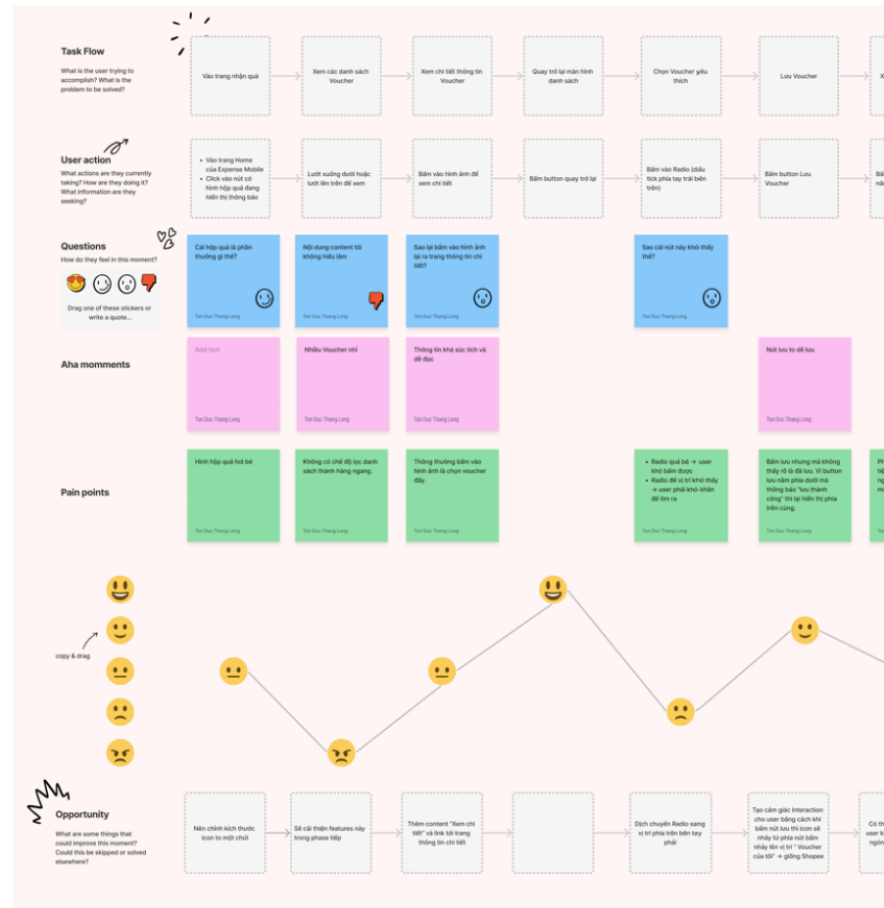
- User Persona
- High-Fidelity Wireframe
- Task Flow
- User Journey
- **User Journey Map**
- Empathy Map

Đầu ra sau khi tiến hành làm Usability Testing là:

- Tìm ra Pain points khi user trải nghiệm trên flow và wireframe
- Tìm ra Opportunity để cải thiện giúp optimize UX
- Mockup đã được improve
- Nghiệm thu để đối chiếu với requirement của các stakeholder

Define **User Journey Map_Advance** theo thứ tự sau:

User Journey Map



- (1) **Task Flow:** Các action mà user cần phải thực hiện để hoàn thành 1 màn hình của User Flow.
- (2) **User Action:** Hành động của User tương tác trên màn hình để hoàn thiện Task Flow đó.

	<p>(3) Question:</p> <p>(4) Emotion (aha moments): Cảm xúc của người dùng ở bước thực hiện đó ra sao</p> <p>(5) Pain Points: Từ đó tìm ra Pain Points ở từng bước (nếu có).</p> <p>(6) Oppotunity: Khắc phục Pain Points cho MVP tiếp theo.</p>	
--	---	--

I. Các bước để tìm ra Feature Set

1. Phỏng vấn định tính để tìm insight của Khách hàng

- Job step để đạt được Gain
- Pain Point trong từng Job step là gì?
- Outcome

2. Phỏng vấn định lượng

- Sau khi tìm ra được outcome (10-15 outcome trở lên) -> validate các Outcomes đó trên nhiều người khác nhau bằng survey theo 2 tiêu chí: Important và Satisfaction.
- Tính điểm Opportunity Score nhằm xác định Underserved Needs.
- Có thể lấy ra 20% outcome đứng đầu đạt 80% điểm cho vào chart 4 vùng (Dan Olsen) để xác định cái nào nên làm trước -> Làm trước Outcome ở vùng **Opportunity**

3. Define Feature set để đạt được Outcome

- Brainstorming: List Job Step - Pain Point - Outcome -> Tìm ra feature để đi được từ Job Step vượt Pain Point tới được Outcome.
- Sau khi có Feature set đưa vào Kanol model để xác định:
 - **Must have**
 - Performance
 - Delighter
- Với MVP thì luôn phải làm các tính năng Must Have trước
- Từ Feature -> breakdown nhỏ ra các Functions tương ứng/ Feature nhỏ hơn trong Feature lớn.
- Sao đó tính điểm ROI (Return/Investment)
 - Return: CEO chấm, Investment_ số H để hoàn thành -> DEV chấm. (story point)
 - Cái nào điểm cao thì sẽ đưa vào làm trước.

II. UX

1. Xác định Wireflow

- User Journey -> **UserFlow** -> Wireflow
- Lưu ý:
 - Sang Level học thêm về Workflow: tương tác của người nhiều role trong một hệ thống.
 - Level 1 học về UserFlow > cần define được Userflow của từng role khác nhau.

2. Chi tiết các màn hình trong Wireflow