Platform Design (III)

V. 3 Factors for Platform success

1. Killer Application

- ✓ Awesome application, function or product/service
  (e.g.) MS Office, KakaoTalk (free message delivery), Naver (search function)
- ✓ Is our platform powerful, differentiated and attractive?
  Considerable value and/or addictive fun

2. Network Effect

- ✓ User demand is influenced by other users
- ✓ The more users you use, the greater the value
- ✓ When a certain point is reached, the product is selected by number rather than quality (Lock-in effect)
  (Example) SNS, facebook

3. Loyalty

- ✓ Make users stay longer
- ✓ How many loyal users do you have?
  (E.g. Amazon, Google, Apple, etc.)

V. 3 Factors for Platform success

1. 킬러 앱 (Killer Application)

- → 끝내주는 응용 프로그램, 기능이나 상품/서비스
- → (예) MS Office, 카카오톡 (무료메시지 전달), 네이버 (검색기능)
- → 우리 플랫폼은 강력하고 차별적이고 매력적인가 ?
- → 상당한 가치 and/or 중독적 재미

2. 네트워크 효과 (Network Effect)

- → 사용자의 수요가 다른 사용자에 의해 영향을 받음
- → 많은 사용자가 사용 할 수록 그 가치가 높아짐
- → 특정시점에 도달하면 품질보단 숫자로 제품이 선택됨 (Lock-in effect)
- → (예) SNS, Facebook

3. 로열티 (Loyalty, 충성도)

- → 사용자들을 오래 머물게 함
- → 충성스런 사용자를 얼마나 갖고 있는가 ?
- → (예) 아마존, 구글, 애플 등

VI. Design Principles (1)

1. Design Principle 1: End-to-End
   - ✓ Activities that are not key but important to a specific user should be located at the end of the network, not at the heart
   - ✓ Application-specific features are at the top of the platform; only the critical features that have the greatest impact on ONLY apps are on the core platform
   - ✓ Run speed and efficiency
   - ✓ The ecosystem of simple core platforms is rapidly evolving
   - ✓ (Example) Amazon Web Service (AWS) vs MS Vista
2. Design Principle 2: Modularity
   - ✓ Strategies for building complex products and processes efficiently (independent designs or functions are integrated)
   - ✓ Clear purpose/direction (cost reduction-development of various products)
   - ✓ Modularity and standard setting through internal and external environment and data analysis
3. Design principle 3: Re-Architecturing

VI. 디자인 원칙 (1)

1. **디자인 원칙 1: 단대단 (End-to-End)**
   - → 핵심은 아니지만 특정사용자에게 중요한 활동은 네트워크의 중심부가 아닌 종단에 위치해야함
   - → 응용 프로그램에 특화된 기능은 플랫폼 맨 위에 위치, ONLY 앱에 가장 많은 영향을 주는 중요한 기능들만 핵심 플랫폼에 위치
   - → 실행 속도와 효율화
   - → 단순한 핵심 플랫폼의 생태계가 빠르게 진화
   - → (예) AWS (Amazon Web Service) vs MS Vista
2. **디자인 원칙 2 : 모듈화 (Modularity)**
   - → **복잡한 제품과 프로세스를 효율적 구축하는 전략 (독립적 설계이나 기능은 통합됨)**
   - → **명확한 목적성/방향성 (비용절감 ? 다양한 제품 개발 ?)**
   - → **내외부 환경과 데이터 분석을 통한 모듈화 및 표준 설정**
3. **디자인 원칙 3 : 재설계(Re-Architecturing)**
   - → 시스템을 모듈화 구조로 완전히 변경

VI. Design principle (2)

Design to create new interactions

❑ Successful Platform

→ Add new interactions on top of core interactions

→ Uber and Lyft → Uberpool & Lyft Line

→ LinkedIn → Discussion Posts → Careers + Related Jobs

→ Changing the value unit exchanged between existing users (linked discussion posting)

→ Attract new categories of users (Linked: Jobs + Advertisers)

→ New type of value unit exchange (Uber + lift)

→ Classify / Organize members of existing groups for new categories (Industrial Leader at LinkedIn)

VI. 디자인 원칙 (2)

새로운 상호작용이 창출되도록 디자인하라

❑ 성공적인 플랫폼

→ 핵심 상호작용 위에 새로운 상호작용을 추가

→ 우버(Uber)와 리프트(Lyft) → Uberpool & Lyft Line

→ 링크드인 → 토론 포스팅 → 채용 + 관련 직종 광고

→ 기존 사용자들 사이에 교환되는 가치 단위 변경 (링크드인 토론 포스팅)

→ 새로운 범주의 사용자를 끌어들인다 (링크드인 : 채용전문가 + 광고업체)

→ 새로운 유형의 가치 단위 교환 (우버+ 리프트)

→ 새로운 범주를 위해 기존 그룹의 회원을 분류/정리 (링크드인의 업계 리더)

VII. Platform Profit/revenue method

1. **Transaction fee charged**
   - ✓ **Uber, Airbnb**
2. **Charge users for enhanced accessibility**
   - ✓ **Yelp's Restaurant Ad Location, Dating Platform (Advanced Information)**
3. **Charges for third party producers in exchange for platform community access**
   - ✓ **Dribble (for designers: job postings for designers), LinkedIn (for Recruiters)**
4. **Service fee charged in exchange for curation enhancement**
   - ✓ **Education Platform Skill Share: Curriculum of high quality course**

VII. Platform revenue method

1. 거래 수수료 부과

   → 우버, 에어비앤비 등

2. 접근성 강화에 대한 대가로 사용자들에게 요금 부과

   → 옐프의 레스토랑 광고 위치, 데이팅 플랫폼 (고급정보)

3. 서드 파티 생산자에게 플랫폼 커뮤니티 접근 대가로 요금 부과

   → 드리블 (디자이너용: 디자이너 구인광고), 링크드인 (채용담당자용)

4. 큐레이션 강화에 대한 대가로 서비스 사용료 부과

   → 교육플랫폼 스킬셰어 : 양질의 강좌를 큐레이션 한 후 수강료 부과

VII. Platform Profit/revenue method(attention point)

1. If possible, don't charge for the value you previously enjoyed for free
2. Do not weaken access to the values users are accustomed to
3. Focus on creating new and additional value to justify these changes as you transition from free to paid
4. Consider potential monetization strategies when designing your platform from the beginning (scratch)

VII. Platform Profit/revenue method(유의점)

1. 가능하면 사용자가 이전에 무료로 누렸던 가치에 요금을 부과하지 마라

2. 사용자들이 익숙하게 느꼈던 가치에 대한 접근성을 약화시키지 마라

3. 무료에서 유료로 전환할 때 이러한 변화를 정당화할 수 있는 새롭고 추가적인 가치 창출에 집중하라

4. 처음부터 플랫폼을 설계할 때 잠재적인 수익 창출 전략을 고려하라

VIII. 3 steps for new platform business

Step 1: 전략 (Strategy)

→ Clarify goals to be oriented and draw business models to realize them

→ Clarify your dreams (ideal goal) → Guess what is the outcome?

→ "New business"? "Planning for new business"?

Step 2: 작전(Operation)

→ Create a business process, an individual project to realize a strategy

→ How does it process and how does it produce results? (Pursuing IT Potential)

Step 3: 전술 (Tactics)

→ Create convenient usage, a means or tool for working on a project

→ IT as a tool → What's the best way to get the job right? What development tools should I use?

VIII. 새로운 플랫폼비즈니스를 위한 3 스텝

Step 1: 전략 (Strategy)

→ 지향해야 할 목표를 분명히 하고, 그것을 실현하기 위한 비즈니스 모델을 그린다

→ 꿈꾸는 모습을 명확히 한다 (이상적인 형태) → 결과가 어떻게 되어 있을 지 ?

→ "신규사업"? "신규사업 계획"?

Step 2: 작전(Operation)

→ 전략을 실현하기 위한 개별 프로젝트인 비즈니스 프로세스를 만든다

→ 어떤 절차로 어떻게 처리해서 어떤 방법으로 결과를 낼까 ? (IT 가능성 추구)

Step 3: 전술 (Tactics)

→ 프로젝트를 수행하기 위한 수단이나 도구인 편리한 사용법을 만든다

→ 도구로서의 IT → 앞으로 실행하려고 하는 작전에 알맞은 수단으로서 최적인 것은 무엇인가 ? 어떤 개발 툴을 사용하면 좋은가 ?