



**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

*Nhóm chuyên môn Nhập môn Công nghệ phần mềm*

# NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

**Các vấn đề trong Công nghệ phần mềm**



1. Xác định đối tượng của phần mềm
2. Cân bằng giữa tính năng, chi phí và thời gian
3. Rủi ro của dự án
4. Một số vấn đề khác

*Sau bài học này, người học có thể biết được*

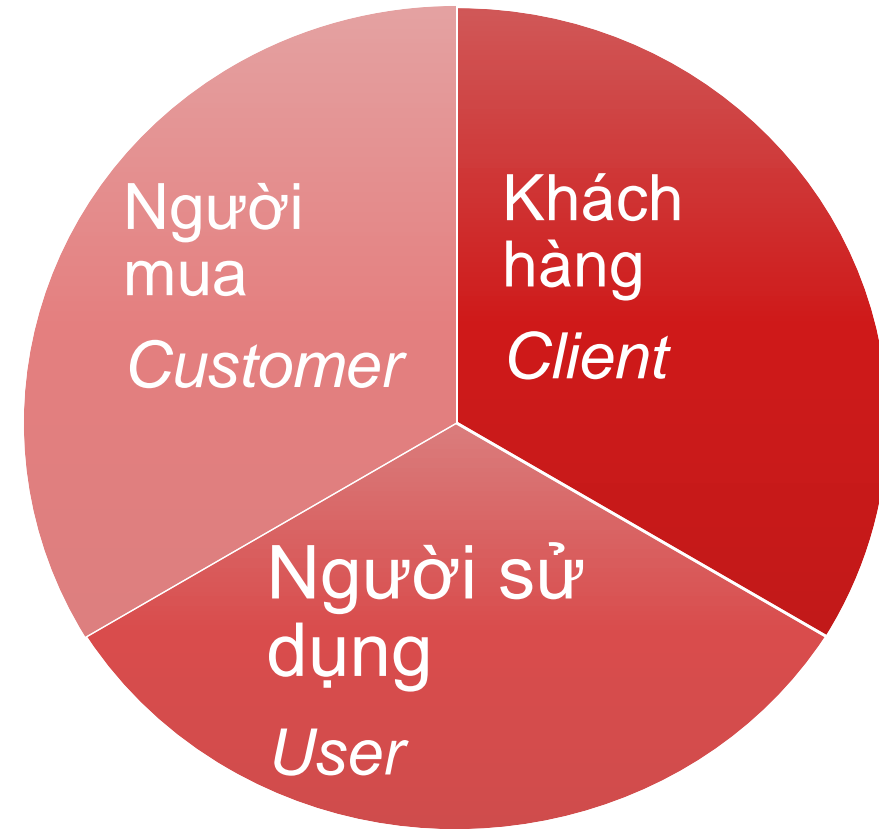
1. Các vấn đề trong việc xác định được **đối tượng** hướng đến của phần mềm
2. Vấn đề về việc **cân bằng** giữa **tính năng**, **chi phí** và **thời gian** của dự án phần mềm
3. Các **rủi ro** có thể xảy ra trong quá trình phát triển dự án
4. Và một số vấn đề khác

## 1. Xác định đối tượng của phần mềm

- 2. Cân bằng giữa tính năng, chi phí và thời gian
- 3. Rủi ro của dự án
- 4. Một số vấn đề khác

# 1. XÁC ĐỊNH ĐỐI TƯỢNG CỦA PHẦN MỀM

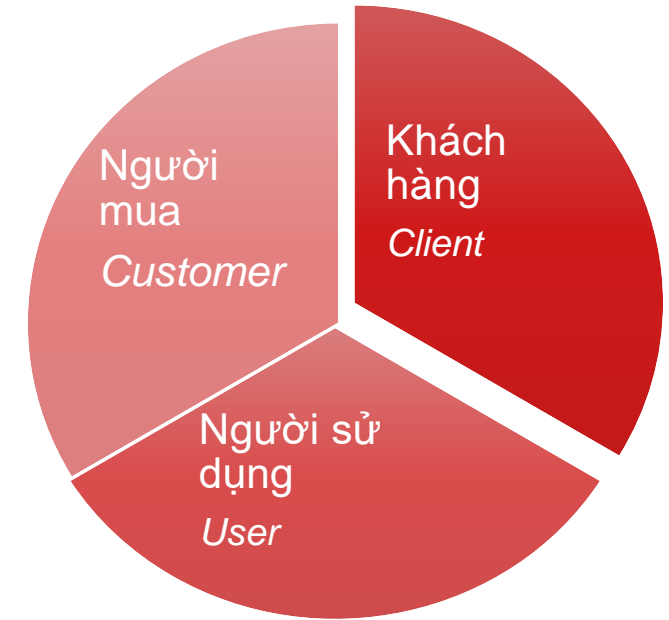
- Phần mềm được phát triển dành cho những đối tượng nào?
  - Khách hàng (**Client**)
  - Người mua (**Customer**)
  - Người sử dụng (**User**)



# 1. XÁC ĐỊNH ĐỐI TƯỢNG CỦA PHẦN MỀM

## ▪ Khách hàng (Client)

- Một người (hoặc nhóm người) mà nhóm phát triển phần mềm tạo ra phần mềm cho họ
- Khách hàng cung cấp các tài nguyên (vd. tiền) và mong đợi nhận lại ở sản phẩm
- Thành công trong công việc của khách hàng có thể phụ thuộc vào sự thành công của dự án phần mềm

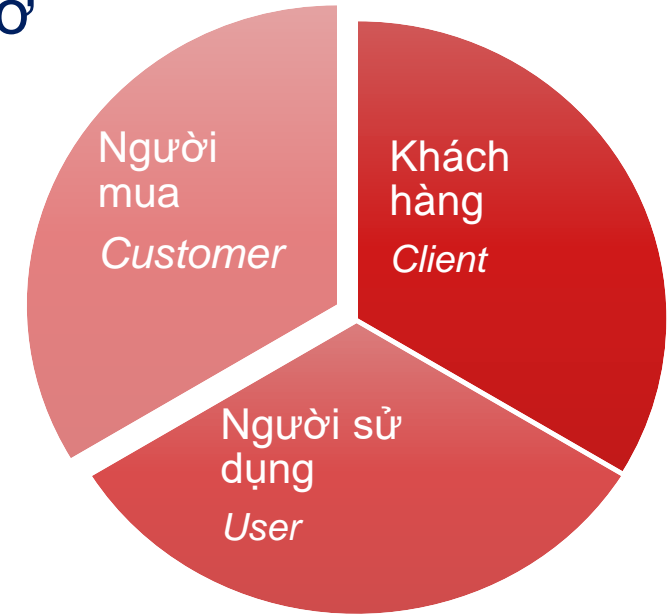


# 1. XÁC ĐỊNH ĐỐI TƯỢNG CỦA PHẦN MỀM

---

## ▪ Người mua (Customer)

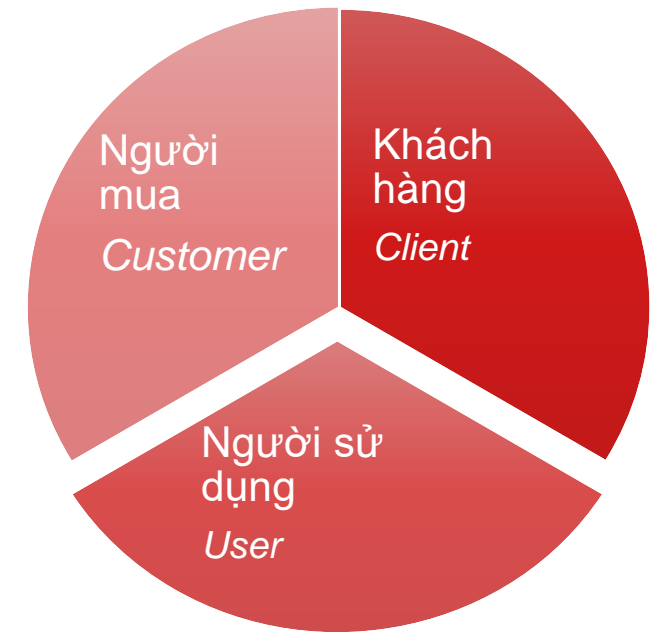
- Là người mua phần mềm hoặc lựa chọn nó để cơ quan/doanh nghiệp sử dụng
- *Khách hàng (client) có phải là người mua (customer) không?*



# 1. XÁC ĐỊNH ĐỐI TƯỢNG CỦA PHẦN MỀM

## ▪ Người sử dụng (User)

- Người dùng, hay người dùng cuối (end-user) là người thực sự sử dụng phần mềm.
- Với phần mềm cá nhân, người dùng và người mua (customer) có thể giống nhau.
- Trong các tổ chức, người mua và người dùng thường khác nhau.



Sự hài lòng của khách hàng là thước đo thành công quan trọng nhất trong dự án phần mềm!



1. Xác định đối tượng của phần mềm

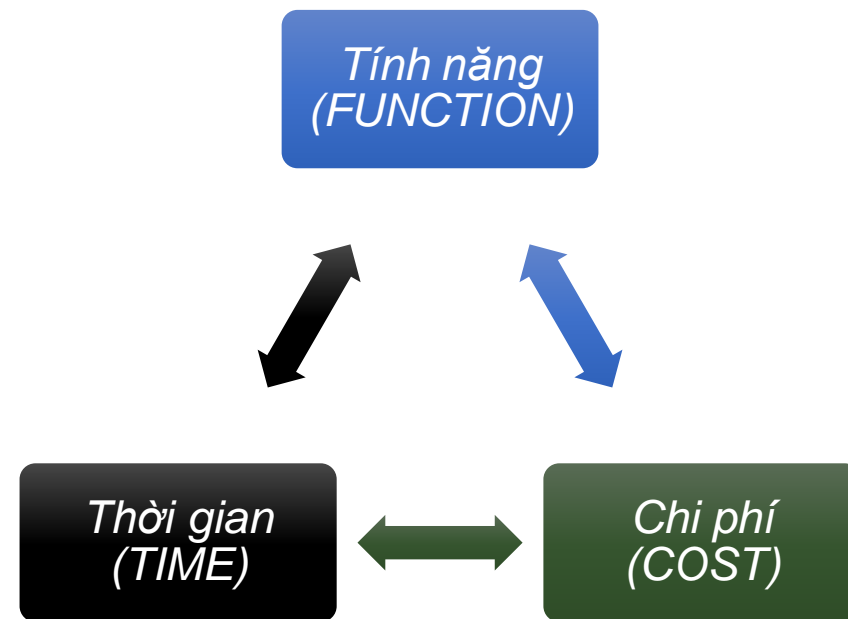
**2. Cân bằng giữa tính năng, chi phí và thời gian**

3. Rủi ro của dự án

4. Một số vấn đề khác

## 2. CÂN BẰNG GIỮA TÍNH NĂNG, CHI PHÍ VÀ THỜI GIAN

- Việc phát triển một dự án phần mềm hướng đến:
  - Sản phẩm hoạt động như mong đợi (**tính năng**)
  - Dưới ngân sách (**chi phí**)
  - Giao hàng đúng hoặc trước hạn (**thời gian**)



- Thực tế:
    - Mọi chức năng bổ sung sẽ tăng thêm chi phí cho việc phát triển, thử nghiệm, bảo trì, v.v.
- Cân bằng giữa chức năng/chi phí/thời gian

1. Xác định đối tượng của phần mềm
2. Cân bằng giữa tính năng, chi phí và thời gian

## **3. Rủi ro của dự án**

4. Một số vấn đề khác

# 3. RỦI RO CỦA DỰ ÁN

---

- Rủi ro
  - Phần lớn chức năng của phần mềm bị **lãng phí** (~50% không bao giờ được sử dụng)
  - Nhiều dự án phần mềm thất bại vì người phát triển phần mềm xây dựng **sai** phần mềm
  - ...
- Nhóm phát triển phần mềm phải:
  - Hiểu những gì khách hàng mong đợi ở phần mềm
  - Hiểu những gì tổ chức của khách hàng mong đợi ở khách hàng
  - Hiểu những gì khách hàng và người dùng mong đợi ở phần mềm

# 3. RỦI RO CỦA DỰ ÁN

---

- Giảm thiểu rủi ro
  - Nghiên cứu tiền khả thi (có nên bắt đầu dự án không?)
  - Tách biệt các yêu cầu (những gì khách hàng muốn) ra khỏi thiết kế (cách nhà phát triển đáp ứng các yêu cầu)
  - Các mốc quan trọng - milestone (cách nhà phát triển báo cáo hoặc chứng minh tiến độ cho khách hàng) và phát hành - release
  - Kiểm thử của người dùng và kiểm thử chấp nhận (cách khách hàng kiểm tra xem phần mềm có đáp ứng yêu cầu không)
  - Bàn giao (đảm bảo khách hàng nhận được gói sản phẩm có thể được vận hành và hỗ trợ trong thời gian dài)

# 3. RỦI RO CỦA DỰ ÁN

---

- Giảm thiểu rủi ro
  - Minh bạch: Người chịu trách nhiệm biết những gì đang diễn ra
  - Các vấn đề (do quản lý nhận ra) phải dựa vào người khác để báo cáo tiến độ hoặc khó khăn
- Vấn đề: người phát triển phần mềm thường
  - Không giỏi đánh giá tiến độ
  - Thường lạc quan về sự tiến triển của dự án
  - Cho rằng việc báo cáo là lãng phí thời gian
  - ...

1. Xác định đối tượng của phần mềm
2. Cân bằng giữa tính năng, chi phí và thời gian
3. Rủi ro của dự án

## **4. Một số vấn đề khác**

## 4. MỘT SỐ VẤN ĐỀ KHÁC

---

- Không có phương pháp **mô tả rõ ràng yêu cầu** của khách hàng
  - Phát sinh vấn đề sau khi bàn giao
- Với những phần mềm quy mô lớn, tư liệu đặc tả **cứng nhắc**
  - Khó đáp ứng nhu cầu thay đổi của người dùng
- Phương pháp luận thiết kế **không nhất quán**
  - Thiết kế theo cách riêng → giảm chất lượng phần mềm
- Không có chuẩn về việc tạo **tài liệu quy trình** sản xuất phần mềm
  - Đặc tả không rõ ràng → giảm chất lượng phần mềm



## 4. MỘT SỐ VẤN ĐỀ KHÁC

---

- Không kiểm thử **tính đúng đắn** của phần mềm ở từng giai đoạn mà chỉ kiểm thử ở giai đoạn cuối
  - Thường bàn giao sản phẩm không đúng hạn
- Không đề cao quá trình thiết kế
  - Giảm chất lượng phần mềm
- Coi thường việc tái sử dụng phần mềm
- Phần lớn các thao tác trong quy trình phát triển phần mềm do con người thực hiện
  - Giảm năng suất lao động

## 4. MỘT SỐ VẤN ĐỀ KHÁC

---

- Không chứng minh được **tính đúng đắn** của phần mềm
  - Giảm độ tin cậy của phần mềm
- Chuẩn về một phần mềm tốt không thể đo được một cách **định lượng**
  - Không thể đánh giá được một hệ thống đúng đắn hay không
- Đầu tư số lượng lớn nhân lực vào bảo trì
  - Giảm hiệu suất lao động của nhân viên

1. Bài học đã giới thiệu cho người học một số các **vấn đề** cần phải được quan tâm trong công nghệ phần mềm, như xác định **đối tượng** phần mềm; **cân bằng** giữa tính năng, chi phí và thời gian phát triển phần mềm, các **rủi ro** trong phát triển phần mềm và một số vấn đề khác.
2. Tiếp sau bài này, người học sẽ tìm hiểu các khái niệm về **vòng đời** phần mềm và các **quy trình phát triển phần mềm**.

# NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

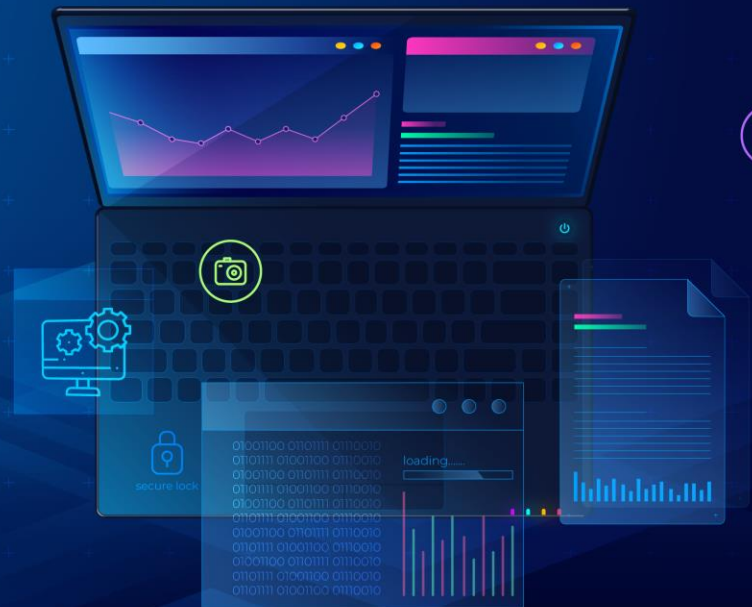
## Các vấn đề trong Công nghệ phần mềm

Biên soạn:

TS. Trịnh Thành Trung

Trình bày:

TS. Trịnh Thành Trung



# NHẬP MÔN CÔNG NGHỆ PHẦN MỀM

*Bài học tiếp theo:*

## Tổng quan vòng đời phần mềm

**Tài liệu tham khảo:**

- [1] R. Pressman, Software Engineering: A Practitioner's Approach. 8th Ed., McGraw-Hill, 2016.
- [2] I. Sommerville, Software Engineering. 10th Ed., AddisonWesley, 2017.
- [3] Pankaj Jalote, An Integrated Approach to Software Engineering, 3rd Ed., Springer.
- [4] Shari Lawrence Pleege, Joanne M. Atlee, Software Engineering theory and practice. 4th Ed., Pearson, 2009