MTAT.03.306
Requirements Engineering

Introduction to RE

Based on
Prof. Steve Easterbrook, Requirements engineering course,
University of Toronto

MTAT.03.306

需求工程

稀土简介

基于多伦多大学需求工程课程 Steve Easterbrook 教授

What is a System?

什么是系统?

Types of System

- Natural Systems
  - E.g. ecosystems, weather, water cycle, the human body, bee colony,…
  - Usually perceived as hard systems
- Abstract Systems
  - E.g. set of mathematical equations, computer programs,…
  - Interesting property: system and description are the same thing
- Symbol Systems
  - E.g. languages, sets of icons, street signs,…
  - Soft because meanings change
- Designed Systems
  - E.g. cars, planes, buildings, freeways, telephones, the internet,…
- Human Activity Systems
  - E.g. businesses, organizations, markets, clubs, …
  - E.g. any designed system when we also include its context of use
    - Similarly for abstract and symbol systems!
- Information Systems
  - Special case of designed systems
    - Part of the design includes the representation of the current state of some human activity system;
    - helps to disseminate information
  - E.g. MIS, banking systems, databases, …
- Control systems
  - Special case of designed systems
    - Designed to control some other system (usually another designed system)
  - E.g. thermostats, autopilots, …

系统类型

- 自然系统
  - 例如生态系统、天气、水循环、人体、蜂群…… ——通常被视为硬系统
- 抽象系统
  - 例如一组数学方程、计算机程序…… - 有趣的属性：系统和描述是同一件事
- 符号系统
  - 例如语言、图标集、街道标志…… ——柔软，因为含义会改变
- 设计系统
  - 例如汽车、飞机、建筑物、高速公路、电话、互联网……
- 人类活动系统
  - 例如企业、组织、市场、俱乐部…… - 例如任何设计的系统，当我们还包括其使用环境时
    - 对于抽象和符号系统也是如此！
- 信息系统
  - 设计系统的特例
    - 部分设计包括表示某些人类活动系统的当前状态；
    - 帮助传播信息
  - 例如MIS 、银行系统、数据库……
- 控制系统
  - 设计系统的特例
    - 设计用于控制其他一些系统（通常是另一个设计的系统）
  - 例如恒温器、自动驾驶仪…… 3

What is software(-intensive) system/ embedded system?

什么是软件（密集型）系统/嵌入式系统?

Software-Intensive Systems

- **Software (on its own) is useless**
  - Software is an abstract description of a set of computations
  - Software only becomes useful when run on some hardware
    - we sometimes take the hardware for granted
  - Software + Hardware = “Computer System”

软件密集型系统

- 软件（本身）是无用的
  - 软件是一组计算的抽象描述 - 软件仅在某些硬件上运行时才变得有用

我们有时认为硬件是理所当然的——软件+硬件=“ 计算机系统”

Software-Intensive Systems

- **Software (on its own) is useless**
  - Software is an abstract description of a set of computations
  - Software only becomes useful when run on some hardware
    - we sometimes take the hardware for granted
  - Software + Hardware = "Computer System"
- **A Computer System (on its own) is useless**
  - Only useful in the context of some human activity that it can support
    - we sometimes take the human context for granted
  - A new computer system will change human activities in significant ways
  - Software + Hardware + Human Activities = "Software-Intensive System"

软件密集型系统

- 软件（本身）是无用的
  - 软件是一组计算的抽象描述 - 软件仅在某些硬件上运行时才变得有用

  我们有时认为硬件是理所当然的——软件+硬件=“ 计算机系统”

- 计算机系统（单独）是没有用的
  - 仅在它可以支持的某些人类活动的背景下有用
    - 我们有时认为人类背景是理所当然的
  - 新的计算机系统将显着改变人类活动 - 软件+硬件+人类活动=“ 软件密集型系统”

Software-Intensive Systems

- **‘Software’ makes many things possible**
  - It is complex and adaptable
  - It can be rapidly changed on-the-fly
  - It turns general-purpose hardware into a huge variety of useful machines

  BUT we need to understand
  - the purpose of the system
  - the context
  - which activities the system should support

软件密集型系统

- “软件”让很多事情成为可能
  - 它复杂且适应性强 - 它可以即时快速更改 - 它将通用硬件转变为各种有用的机器

但我们需要明白

- 系统的目的
- 背景 - 系统应支持哪些活动

Control Systems

Needs to ensure safe control of

Tracks and controls the state of

Subject system

Usage System

Uses

Control system

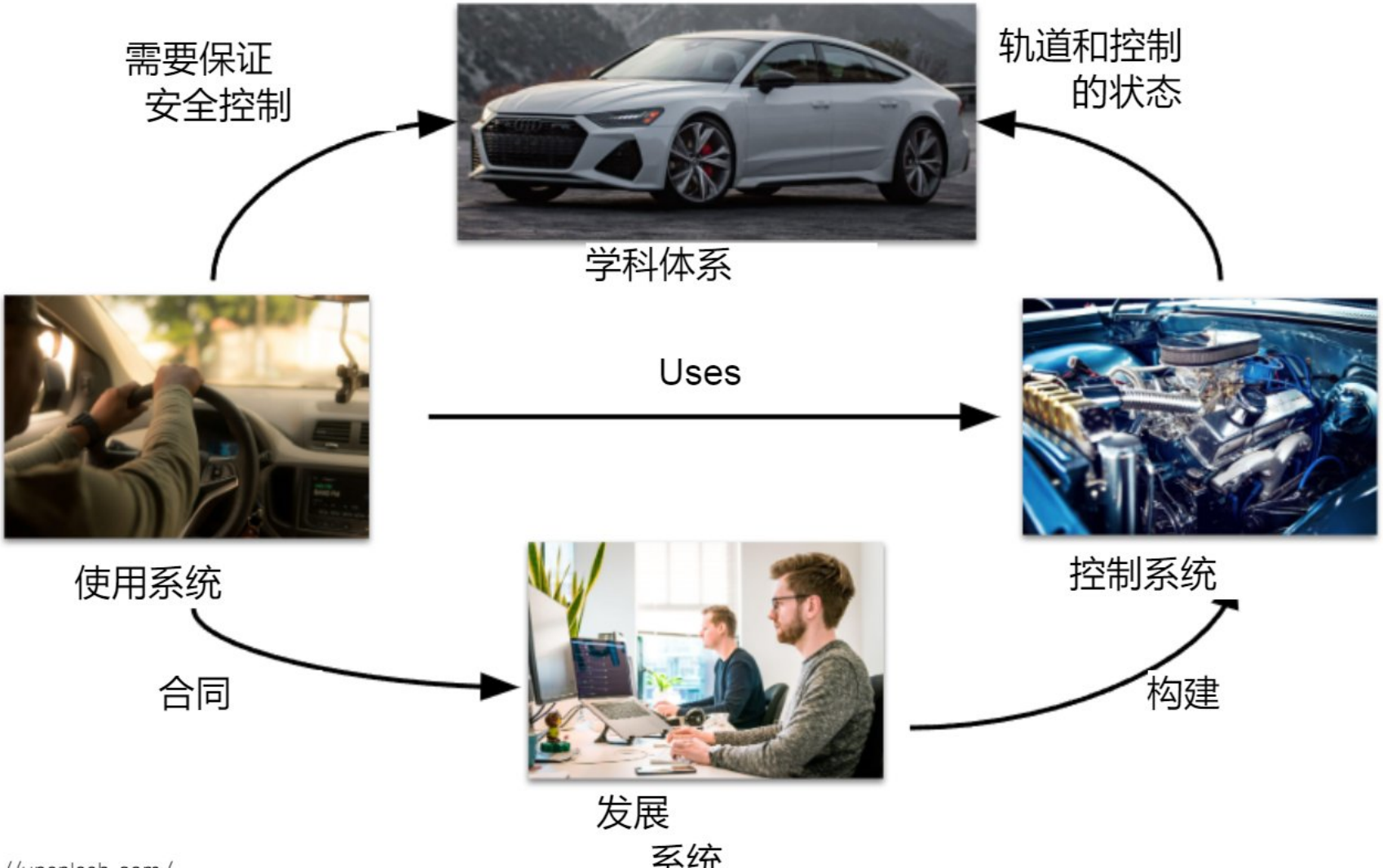
contracts

builds

Development System

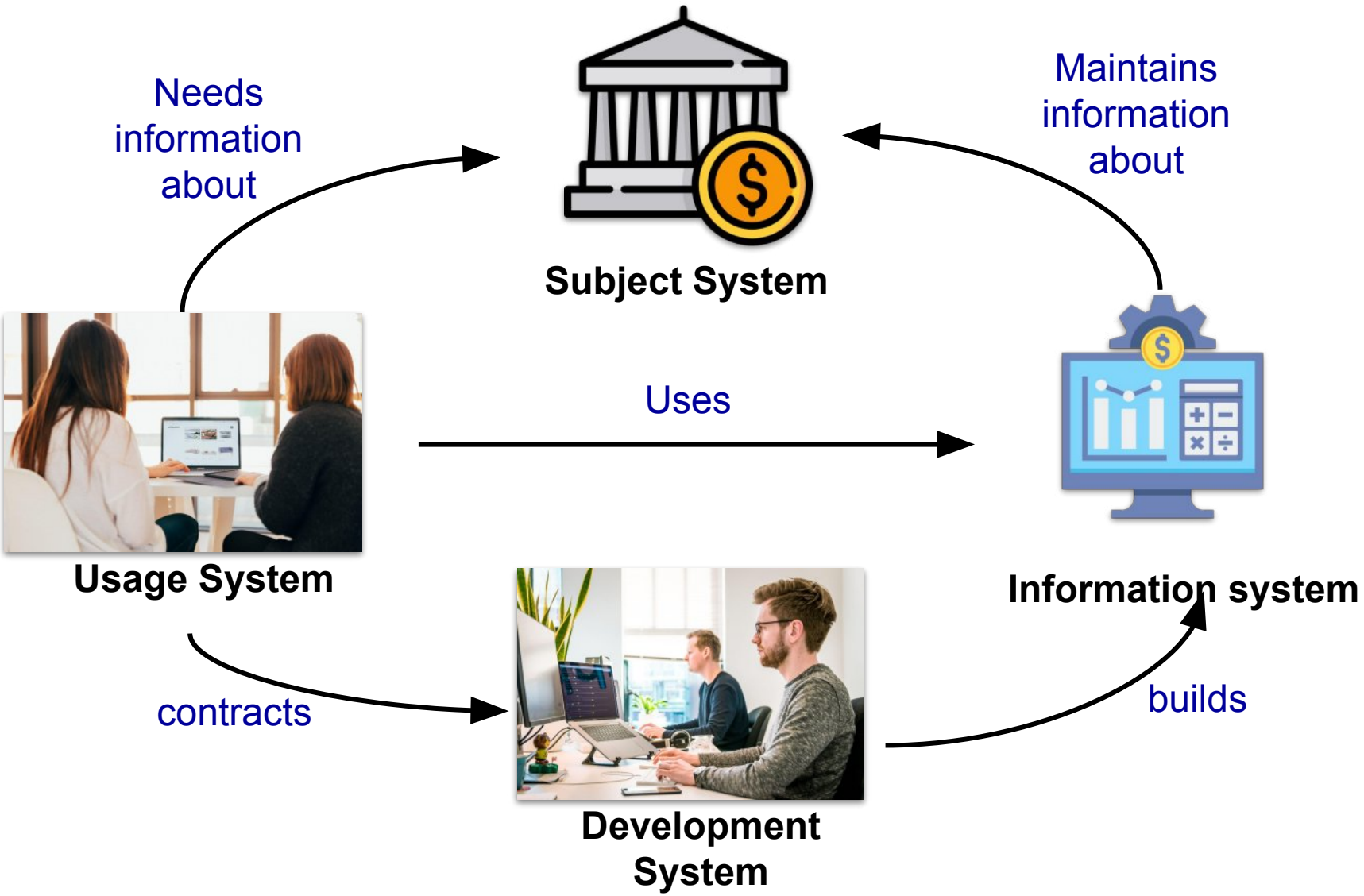
https://unsplash.com/

控制系统



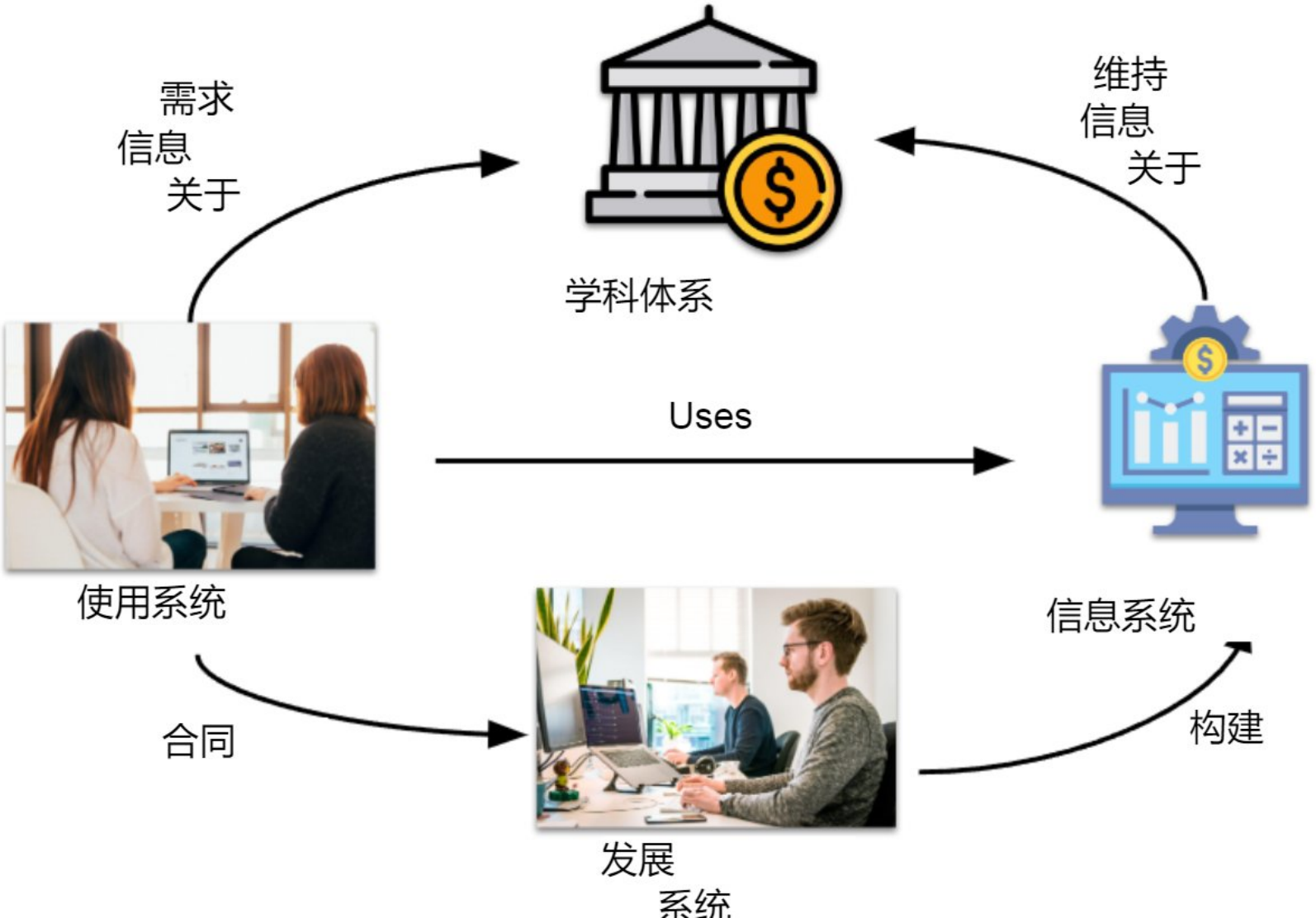
Information Systems

Source: *Adapted from Loucopoulos & Karakostas, 1995, p73*



信息系统

资料来源：改编自 *Loucopoulos* 和 *Karakostas* ，*1995* 年，第 *73* 页



Where are the challenges?

(when building these systems)

挑战在哪里?

(构建这些系统时)

Cost of getting it wrong

	2004	2008	2020
Successful	16%	26%	31%
Challenged	53%	46%	50%
Failure	31%	28%	19%

- Cost of fixing errors
  - Typical development process:

    requirements analysis ⇒ software design ⇒ development ⇒ development testing ⇒ acceptance testing ⇒ operation
  - Errors cost more to fix the longer they are undetected
    - E.g. A requirements error found in testing costs 100 times more than a programming error found in testing

犯错的代价

	200420082020		
成功 16％ 26%	31％ 面临挑战 53%	46% 50% 失败 31%	28% 19％

- 修复错误的成本——典型的开发流程：

需求分析⇒软件设计⇒开发⇒开发测试⇒验收测试⇒运行

– 错误未被发现的时间越长，修复成本就越高
  - 例如测试中发现的需求错误的成本是测试中发现的编程错误的100倍

Cost of getting it wrong

- Causes of project failure
  - Survey of US software projects by the Standish group:

Top 3 success factors:

1) User involvement (Good Place)
2) Executive management support (Good sponsor)
3) Clear statement of requirements (Good team)

Factors leading to failure:

1) Incomplete Requirements
2) Lack of user involvement

…

6) Changing requirements & specs

Most of them are from RE activities and understanding the need

https://www.opendoorerp.com/the-standish-group-report-83-9-of-it-projects-partially-or-completely-fail/

犯错的代价

- 项目失败的原因——Standish 集团对美国软件项目的调查：

三大成功因素：
1) 用户参与（好地方） 2) 执行管理支持（好发起人） 3) 明确的要求说明（好团队）

导致失败的因素：
1）需求不完整
2）缺乏用户参与
… 6) 不断变化的要求和规格

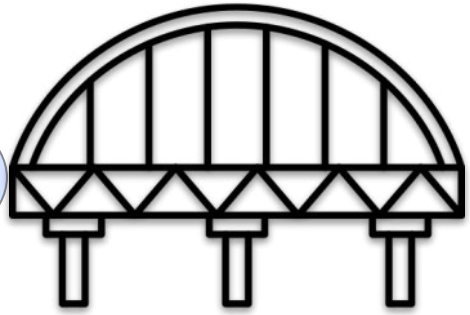
其中大部分来自 RE 活动并了解需求

https://www.opendoorerp .com/the-standish-group -report -83-9-of-it-projects -partially -or-completely -fail/

Where are the challenges?

Application Domain

Machine Domain



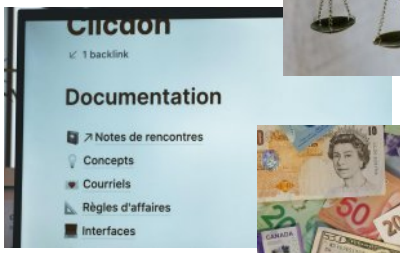
domain properties

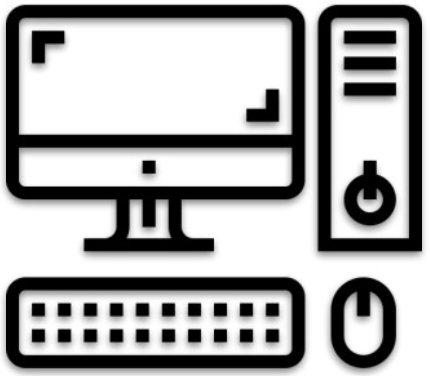
requirements

programs

computers

specification





挑战在哪里?

应用域 机器域

域属性

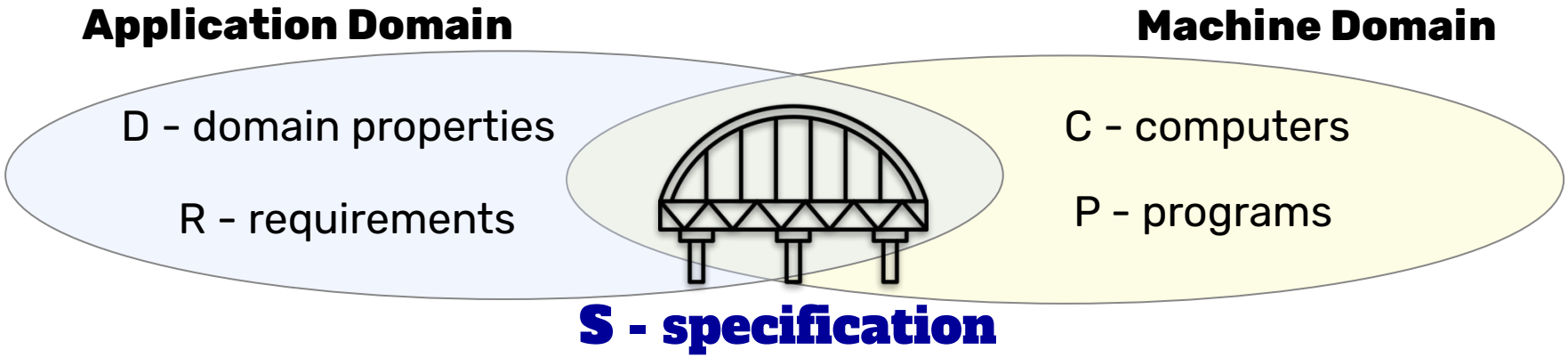
要求

程式

电脑

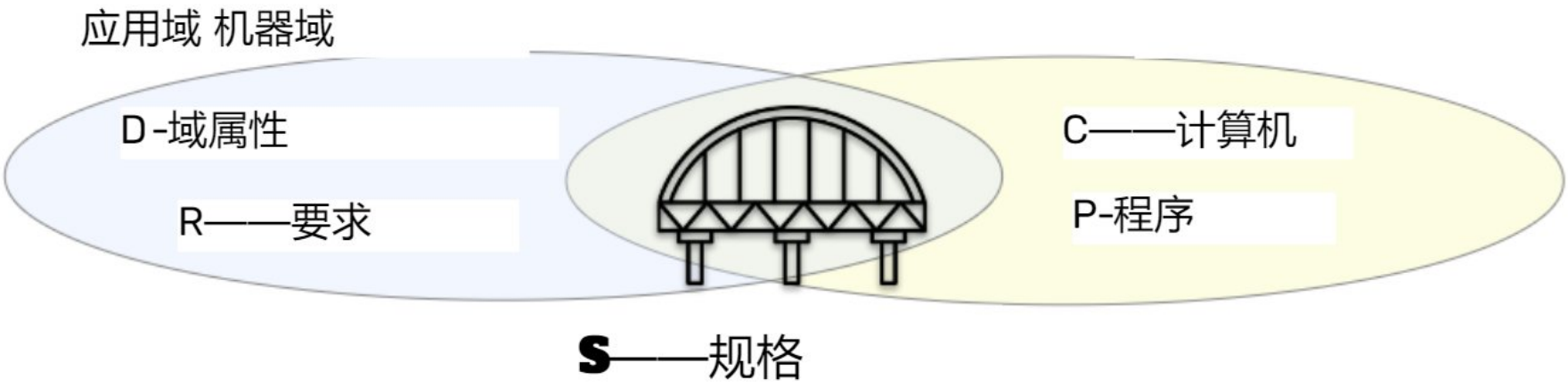
规格

What are requirements?



- **Domain Properties:**
  - things in the application domain that are true whether or not we ever build the proposed system
- **Requirements:**
  - things in the application domain that we wish to be made true by delivering the proposed system
    - Many of which will involve phenomena the machine has no access to
- **A Specification:**
  - is a description of the behaviours that the program must have in order to meet the requirements
    - Can only be written in terms of shared phenomena!

有什么要求?



- 域属性:
  - 无论我们是否构建了所提议的系统，应用程序领域中的事物都是真实的
- 要求:
  - 我们希望通过交付所提议的系统来实现应用领域中的事情
    - 其中许多将涉及机器无法访问的现象
- 一、规格:
  - 是程序为了满足要求而必须具有的行为的描述
    - 只能用共同现象来写！

A requirement is a **usable representation** of a **need**. Requirements focus on understanding what kind of value could be delivered if a requirement is fulfilled

Babook v2 p.15

e.g., “The system enables the sales manager to run a report of the insurances sold over a given time period.”

需求是一个可用的表示

的需要。需求侧重于理解如果满足需求可以交付什么样的价值

例如，“ 该系统使销售经理能够运行给定时间段内销售的保险的报告。”

What are the requirements of the UT students for the study info system (SIS)?

有什么要求

UT 学生的学习信息系统 **(SIS)** 是多少?

What is engineering?

"Engineering is the development of cost-effective solutions to practical problems, through the application of scientific knowledge"

"…Cost-effective…"

- Consideration of design trade-offs, esp. resource usage
- Good enough
- Minimize negative impacts (e.g. environmental and social cost)

"… Solutions …"

- Emphasis on building devices

"… Practical problems …"

- solving problems that matter to people
- improving human life in general through technological advance

"… Application of scientific knowledge …"

- Systematic application of analytical techniques

什么是工程?

“ 工程是通过应用科学知识，为实际问题开发具有成本效益的解决方案”

“ …… 性价比高… …”

– 考虑设计权衡，尤其是。资源利用 – 足够好 – 尽量减少负面影响（例如环境和社会成本）“ … … 解决方案… …”

– 强调构建设备“ … … 实际问题… …”

– 解决与人们息息相关的问题 – 通过技术进步改善人类生活“ … … 科学知识的应用… …”

– 分析技术的系统应用

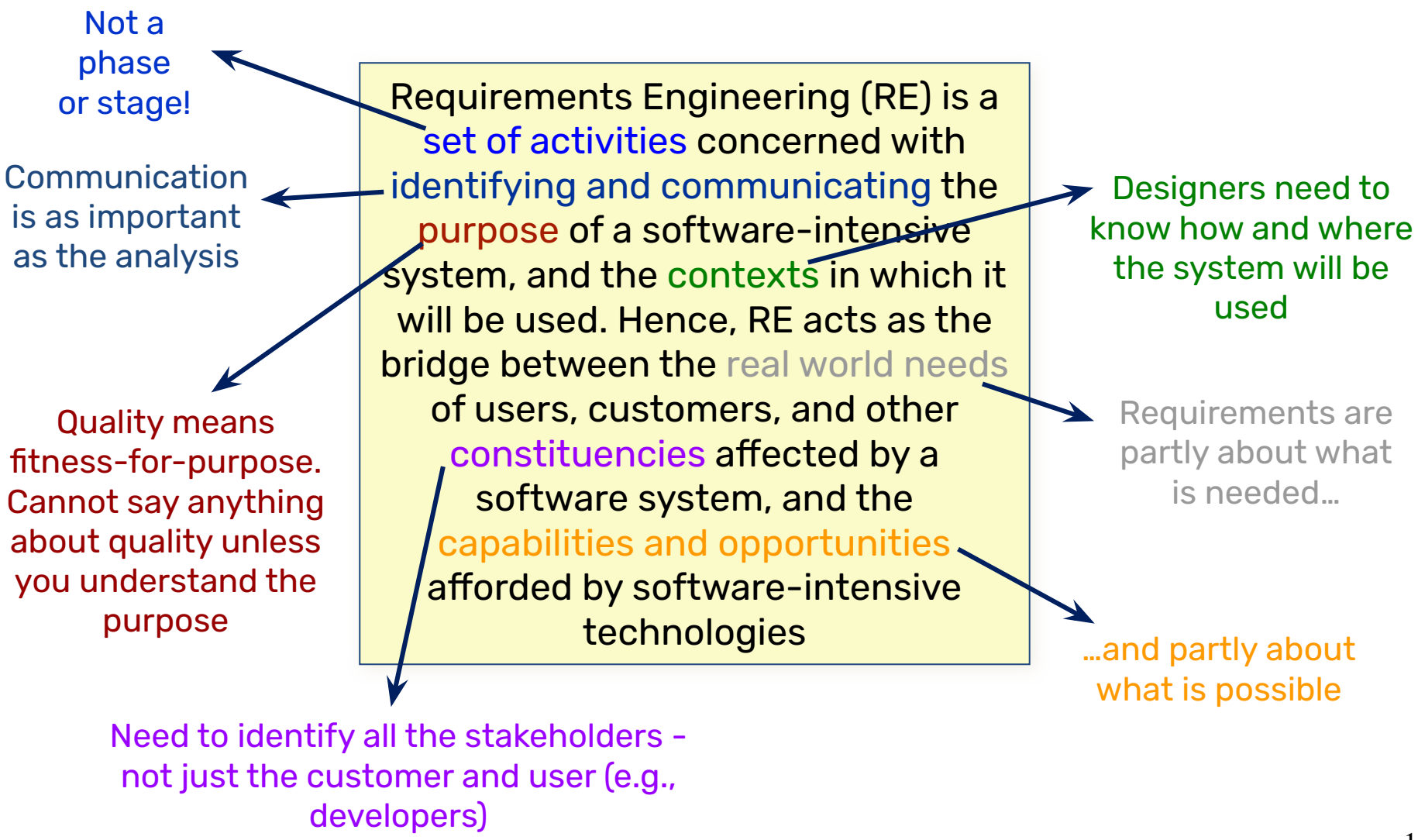
Definition of RE

Requirements Engineering (RE) is a set of activities concerned with identifying and communicating the purpose of a software-intensive system, and the contexts in which it will be used. Hence, RE acts as the bridge between the real world needs of users, customers, and other constituencies affected by a software system, and the capabilities and opportunities afforded by software-intensive technologies

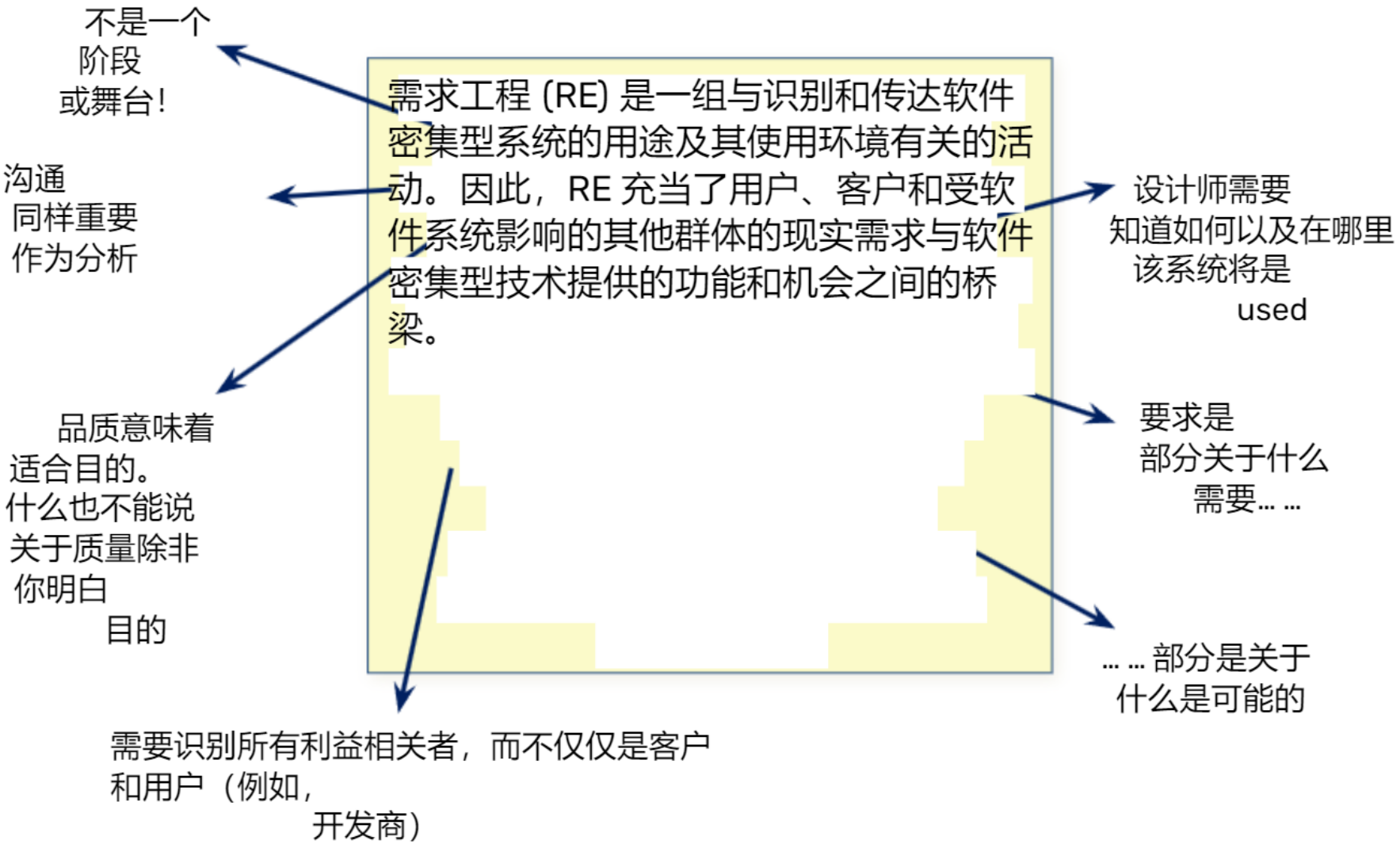
稀土的定义

需求工程 (RE) 是一组与识别和传达软件密集型系统的用途及其使用环境有关的活动。因此，RE 充当了用户、客户和受软件系统影响的其他群体的现实需求与软件密集型技术提供的功能和机会之间的桥梁。

Definition of RE



稀土的定义


不是一个
阶段
或舞台!
沟通
同样重要
作为分析
品质意味着
适合目的。
什么也不能说
关于质量除非
你明白
目的
需求工程 (RE) 是一组与识别和传达软件密集型系统的用途及其使用环境有关的活动。因此，RE 充当了用户、客户和受软件系统影响的其他群体的现实需求与软件密集型技术提供的功能和机会之间的桥梁。
设计师需要
知道如何以及在哪里
该系统将是
used
要求是
部分关于什么
需要… …
… … 部分是关于
什么是可能的
需要识别所有利益相关者，而不仅仅是客户
和用户（例如，
开发商）

Some observations about RE

- **RE is not necessarily a sequential process**
  - Don't have to write the problem statement before the solution statement
    - (Re-)writing a problem statement can be useful at any stage of development
  - RE activities continue throughout the development process
- **The problem statement will be imperfect**
  - RE models are approximations of the world
    - will contain inaccuracies and inconsistencies
    - will omit some information.
    - analysis should reduce the risk that these will cause serious problems…

关于RE的一些观察

- **RE** 不一定是一个顺序过程
  - 不必在解决方案陈述之前写问题陈述
    - （重新）编写问题陈述在开发的任何阶段都是有用的
  - 可再生能源活动在整个开发过程中持续进行
- 问题陈述将不完善
  - RE 模型是世界的近似值
    - 将包含不准确和不一致的内容
    - 会省略一些信息。
    - 分析应该降低这些会导致严重问题的风险.....

Some observations about RE

- **Perfecting a specification may not be cost-effective**
  - Requirements analysis has a cost
  - For different projects, the cost-benefit balance will be different
  - Use principle of ***good enough***
- **Problem statement should never be treated as fixed**
  - Change is inevitable, and therefore must be planned for
  - There should be a way of incorporating changes periodically

关于RE的一些观察

- 完善规范可能不具有成本效益
  - 需求分析是有成本的 – 对于不同的项目，成本收益平衡会不同 – 采用足够好的原则
- 问题陈述绝不应被视为已解决
  - 变化是不可避免的，因此必须进行计划 – 应该有一种定期合并变化的方法

What does requirements engineer do?

有什么要求

工程师做什么?

What does requirements engineer do?

- Works directly w/ stakeholders
- Elicits requirements
- Analyzes requirements
- Documents requirements
- May create functional specifications

有什么要求

工程师做什么?

- 直接与利益相关者合作
- 引出需求
- 分析需求
- 文件要求
- 可以创建功能规范

What does requirements engineer do?

- Explain the concepts, theories, and best practices associated with requirements engineering
- Elicit, negotiate and document software requirements
- Develop major requirements artefacts and use them during the software development projects
- Apply requirements validation techniques
- Manage software requirements, priorities, and trace them

有什么要求

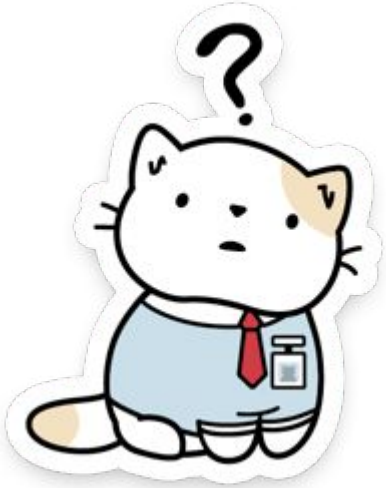
工程师做什么?

- 解释与需求工程相关的概念、理论和最佳实践

- 征求、协商和记录软件需求
- 开发主要需求工件并在软件开发项目中使用它们

- 应用需求验证技术
- 管理软件需求、优先级并跟踪它们

Any questions



任何问题