

社会技术系统

目标

- 解释什么是社会技术系统以及它与基于计算机的系统之间的区别
- 引入紧急系统属性的概念，例如可靠性和安全性
- 解释系统工程和系统采购流程
- 解释为什么系统的组织环境会影响其设计和使用
- 讨论遗留系统以及为什么这些系统对许多企业至关重要

涵盖的主题

- 紧急系统属性
- 系统工程
- 组织、人员和计算机系统
- 遗留系统

什么是系统？

- 相互关联的组件的有目的的集合，它们协同工作以实现某些共同目标。
- 系统可能包括软件、机械、电气和电子硬件，并由人操作。
- 系统组件依赖于其他系统组件
- 系统组件的属性和行为密不可分地交织在一起

系统类别

- 基于计算机的技术系统
 - 包括硬件和软件但操作员和操作过程通常不被视为系统一部分的系统。系统没有自我意识。
- 社会技术系统
 - 系统包括技术系统，但也包括操作流程和使用技术系统并与之交互的人员。社会技术系统受组织政策和规则的约束。

社会技术系统特征

- 紧急属性
 - 依赖于系统组件及其关系的整体系统的属性。
- 非确定性
 - 当呈现相同的输入时，它们并不总是产生相同的输出，因为系统的行为部分依赖于人类操作员。
- 与组织目标的复杂关系
 - 系统支持组织目标的程度不仅仅取决于系统本身。

紧急属性

- 整个系统的属性，而不是可以从系统组件的属性中推导出来的属性
- 紧急属性是系统组件之间关系的结果
- 因此，只有在将组件集成到系统中后才能对其进行评估和测量

涌现属性的例子

紧急财产的类型

- 功能特性
 - 当系统的所有部分协同工作以实现某个目标时，就会出现这些问题。例如，自行车一旦由其部件组装而成，就具有作为运输设备的功能特性。
- 非功能性涌现属性
 - 例如可靠性、性能、安全和保障。这些与系统在其操作环境中的行为有关。它们对于基于计算机的系统通常很重要，因为未能在这些属性中达到某个最低定义级别可能会使系统无法使用。

系统可靠性工程

- 由于组件的相互依赖性，故障可以通过系统传播。
- 系统故障经常由于组件之间不可预见的相互关系而发生。
- 预测所有可能的组件关系可能是不可能的。
- 软件可靠性测量可能会给出系统可靠性的错误描述。

对可靠性的影响

- *硬件可靠性*
 - 硬件组件发生故障的概率是多少？修复该组件需要多长时间？
- *软件可靠性*
 - 软件组件产生错误输出的可能性有多大。软件故障通常不同于硬件故障，因为软件不会磨损。
- *操作员可靠性*
 - 系统操作员出错的可能性有多大？

可靠性关系

- 硬件故障会产生超出软件预期输入范围的杂散信号。
- 软件错误会导致激活警报，从而导致操作员压力并导致操作员错误。
- 系统安装的环境会影响其可靠性。

“不应”属性

- 可以测量性能和可靠性等属性。

- 但是，某些属性是系统不应显示的属性
 - 安全 - 系统不应以不安全的方式运行；
 - 安全性 - 系统不应允许未经授权的使用。
- 测量或评估这些属性非常困难。

系统工程

- 指定、设计、实施、验证、部署和维护社会技术系统。
- 关注系统提供的服务，对其构建和运行的约束以及它的使用方式。

系统工程过程

- 通常遵循“瀑布”模型，因为需要并行开发系统的不同部分
 - 阶段之间的迭代空间很小，因为硬件更改非常昂贵。软件可能必须补偿硬件问题。
- 不可避免地涉及必须一起工作的来自不同学科的工程师
 - 这里有很大的误解空间。不同的学科使用不同的词汇，需要进行大量协商。工程师可能有个人议程要完成。

系统工程过程

跨学科参与

系统需求定义

- 在这个阶段定义的三种类型的需求
 - 抽象的功能需求。系统功能以抽象方式定义；
 - 系统属性。一般定义系统的非功能性需求；
 - 不受欢迎的特性。指定了不可接受的系统行为。
- 还应定义系统的总体组织目标。

系统目标

- 应该定义为特定环境采购系统的原因。
- 功能目标
 - 为建筑物提供火灾和入侵者警报系统，该系统将对火灾或未经授权的入侵提供内部和外部警告。
- 组织目标
 - 确保建筑物内进行的工作的正常运作不会因火灾和未经授权的入侵等事件而受到严重干扰。

系统需求问题

- 复杂的系统通常是了解决棘手的问题而开发的
 - 未完全理解的问题；
 - 随着系统的指定而改变。
- 必须预测系统生命周期内的硬件/通信发展。
- 在不了解系统组件结构的情况下，很难定义非功能性需求（尤其是）。

系统设计流程

- 分区要求
 - 将需求组织到相关组中。
- 识别子系统
 - 确定一组可以共同满足系统要求的子系统。
- 为子系统分配需求
 - 集成 COTS 时会导致特定问题。
- 指定子系统功能。
- 定义子系统接口
 - 并行子系统开发的关键活动。

系统设计流程

系统设计问题

- 对硬件、软件和人工组件的需求划分可能涉及大量协商。
- 困难的设计问题通常被认为可以使用软件轻松解决。
- 硬件平台可能不适合软件要求，因此软件必须对此进行补偿。

要求和设计

- 需求工程和系统设计是密不可分的。
- 系统环境和其他系统造成的约束限制了设计选择，因此可能需要使用实际的设计。
- 可能需要初始设计来构建需求。
- 在进行设计时，您会了解更多有关要求的信息。

需求/设计的螺旋模型

系统建模

- 架构模型呈现构成系统的子系统的抽象视图
- 可能包括子系统之间的主要信息流
- 通常以框图形式呈现

- 可以识别模型中不同类型的功能组件

防盗报警系统

子系统说明

空管系统架构

子系统开发

- 通常并行项目开发硬件、软件和通信。
- 可能涉及一些 COTS（商业现货供应）系统采购。
- 实施团队之间缺乏沟通。
- 提出系统更改的官僚主义和缓慢的机制意味着由于需要返工，开发时间表可能会延长。

系统集成

- 将硬件、软件和人员组合在一起形成系统的过程。
- 应该逐步解决，以便一次集成一个子系统。
- 这个阶段通常会发现子系统之间的接口问题。
- 可能是系统组件交付不协调的问题。

系统安装

- 完成后，系统必须安装在客户的环境中
 - 环境假设可能不正确；
 - 可能是人类对引入新系统的抵制；
 - 系统可能必须与替代系统共存一段时间；
 - 可能是物理安装问题（例如布线问题）；
 - 必须确定操作员培训。

系统进化

- 大型系统的使用寿命很长。它们必须不断发展以满足不断变化的需求。
- 进化本质上是昂贵的
 - 必须从技术和业务角度分析变化；
 - 子系统相互作用，因此可能出现意想不到的问题；
 - 原始设计决策很少有理由；
 - 系统结构因更改而损坏。
- 必须维护的现有系统有时称为遗留系统。

系统退役

- 使系统在其使用寿命结束后停止服务。
- 可能需要去除污染环境材料（例如危险化学品）
 - 在系统设计中应通过封装进行规划。
- 可能需要对数据进行重组和转换，以便在其他系统中使用。

组织/人员/系统

- 社会技术系统是旨在帮助实现某些组织或业务目标的组织系统。
- 如果您不了解使用系统的组织环境，则该系统不太可能满足业务及其用户的实际需求。

人力和组织因素

- 流程变更
 - 系统是否需要更改环境中的工作流程？
- 工作变动
 - 系统是否会降低环境中用户的技能或导致他们改变他们的工作方式？
- 组织变革
 - 该制度是否改变了组织中的政治权力结构？

组织流程

- 系统工程的过程与组织采购过程重叠并相互作用。
- 操作流程是将系统用于其预期目的所涉及的流程。对于新系统，必须将这些定义为系统设计的一部分。
- 操作流程应设计为灵活的，不应强制以特定方式进行操作。如果出现问题，人类操作员可以发挥他们的主动性，这一点很重要。

采购/开发流程

系统采购

- 为组织获取系统以满足某些需要
- 采购前通常需要一些系统规范和架构设计
 - 您需要一个规范来签订系统开发合同
 - 该规范可能允许您购买商业现货 (COTS) 系统。几乎总是比从头开发系统便宜
- 大型复杂系统通常由现成的和专门设计的组件组成。这些不同类型组件的采购流程通常是不同的。

系统采购流程

采购问题

- 可能需要修改需求以匹配现成组件的功能。
- 需求规范可能是系统开发合同的一部分。
- 在选择构建系统的承包商后，通常会有一个合同谈判期来同意更改。

承包商和分包商

- 大型硬件/软件系统的采购通常基于一些主要承包商。
- 向其他供应商发出分包合同以供应系统的部件。
- 客户与总承包商联络，不直接与分包商打交道。

承包商/分包商模式

遗留系统

- 使用旧的或过时的技术开发的社会技术系统。
- 对企业运营至关重要，丢弃这些系统通常风险太大
 - 银行客户核算系统；
 - 飞机维修系统。
- 遗留系统限制了新的业务流程并消耗了大量的公司预算。

遗留系统组件

- 硬件 - 可能是过时的大型机硬件。
- 支持软件 - 可能依赖于不再营业的供应商的支持软件。
- 应用软件 - 可能用过时的编程语言编写。
- 应用程序数据 - 通常不完整且不一致。
- 业务流程 - 可能受软件结构和功能的限制。
- 业务策略和规则 - 可能是隐含的并嵌入到系统软件中。

关键点

- 社会技术系统包括计算机硬件、软件和人员，旨在满足某些业务目标。
- 紧急属性是作为整个系统而不是其组成部分的特征的属性。
- 系统工程过程包括规范、设计、开发、集成和测试。系统集成尤为关键。

关键点

- 人和组织因素对社会技术系统的运行有显著影响。
- 系统采购、开发和运行过程之间存在复杂的交互作用。
- 遗留系统是继续提供基本服务的旧系统。
- 遗留系统包括业务流程、应用软件、支持软件和系统硬件。