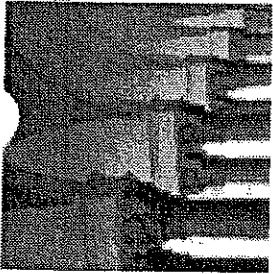


# Objectives

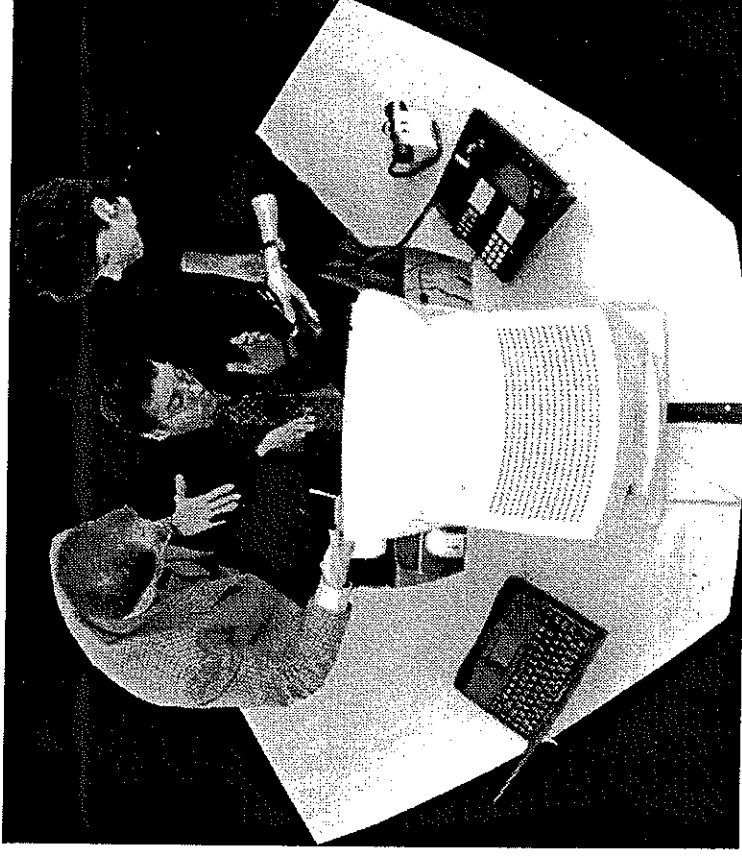
---

- Understand the definition of a project and software
- Understand the role of requirements
- Explain the differences between traditional project management and software project management
- Have an overview of what is involved in managing a software project



# Why is Software Important?

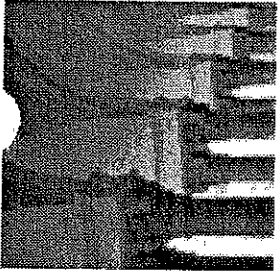
- Almost all industries now depend on software for competitive advantage
  - MRP/ERP
  - Supply Chain Management
  - POS
  - CRM
  - HR
  - Finance
  - Custom applications



# What is a Project?

---

- A project is a temporary endeavor undertaken to create a unique product or service
  - *Temporary* means that every project has a definite beginning and a definite end
  - *Unique* means the product or service is different in some distinguishing way from all of the other products or services



# What is Software?

---

- Computer **programs, procedures, rules, and associated documentation and data**, pertaining to the operation of a computer system

# Why do I need Requirements?

---

- The requirements specification is a complete description of the functionality and behavior of the product
  - Becomes a contract of the product to be built
  - The cost of good requirements gathering and systems analysis is minor compared to the cost of bad requirements
  - Up to 60% of errors originate with requirements and analysis

# What is a Requirement?

---

**A requirement is something the product must do or a quality the product must have**

- **Functional Requirements**

- The things the product must do
- An action the product must take to provide useful functionality

- **Non-Functional Requirements**

- Properties or qualities the product must have

- **Constraints**

- Preferably defined at the beginning of the process
- Global restrictions and limitations that apply to the project and the product

# Functional Requirements

---

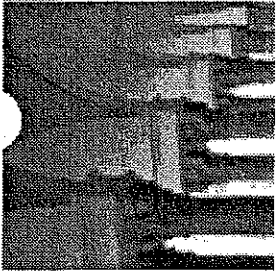
- Specifications of the product's functionality
- Actions the product must take
- Derived from the fundamental purpose of the product
- Not a quality of the product
- Should form a complete and unambiguous description of the product's functionality

# Non-Functional Requirements

---

- Look and feel requirements
- Usability requirements
- Performance requirements
- Operational requirements
- Maintainability requirements
- Security requirements
  - Confidentiality, Availability, Integrity
- Cultural and political requirements
- Legal requirements

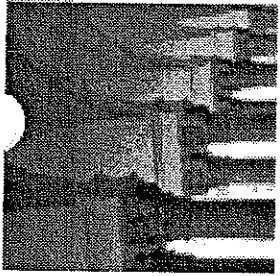




# Constraints

---

- Constraints are issues that have a strong influence on the requirements and outcome for the product and are usually derived from management policy
  - **Design constraints** are pre-existing decisions mandating how the product must look or the technology it must use
  - **Project constraints** are things like budget, schedule, deadlines, etc.



# Managing a Software Project

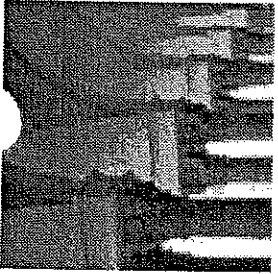
---

- Identify objectives, goals and deliverables
- Select lifecycle
- Develop a project plan
  - WBS, milestones, dependencies
- Assess / review plan
  - Critical path, alternatives
- Implement / execute plan
  - Tracking, reporting, adjustments, risk management
- Deliver product or service
- Post Mortem

# Project Management of Software Projects

---

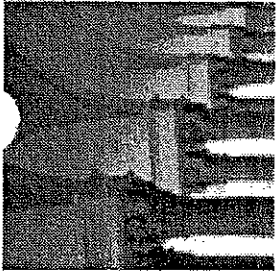
- Harder to quantify what you're doing
- Projects start out with very little defined and understood
- Projects end with well defined deliverables
- Some deliverables differ
  - Intellectual property
  - Binary image
  - Documentation
  - APIs, platform requirements
  - Known (and unknown) limitations



# Software v. Traditional Project Management

---

- Many traditional project management techniques are applicable
  - Statement of work
  - Work breakdown structures
  - Project plans
  - Project tracking
  - Specifications
  - Technical deliverables

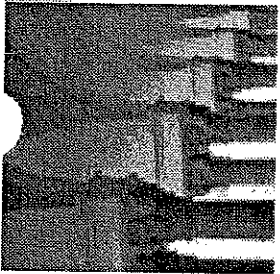


# Software v. Traditional Project Management

---

- Attributes of a good program
  - It works
  - It works according to specifications
  - It is flexible
  - It is ready on time
  - It has no bugs
  - The bugs, which are inevitable, can be fixed quickly
  - It is well documented
  - It executes quickly
  - It makes efficient use of memory

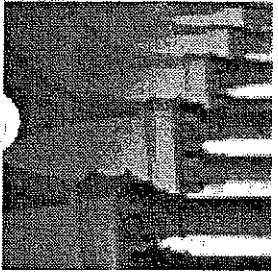
Dr. Edward Yourdon, Techniques of Program Structure and Design, pg 7



# Software Quality

---

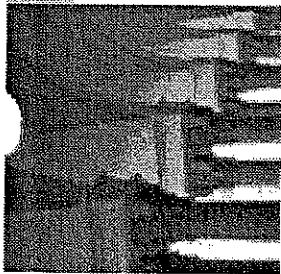
- Satisfy requirements
- Economical
- Efficient, Robust, Reliable
- Easy to Use
- Scalable
- Supports Re-use



# Attributes of Good Software Project Management

---

- Products are delivered on schedule and within cost estimates
- Good product is delivered
- Product defects are known
- Product design reflects user skill sets
- Requirements traceability can be demonstrated



# Exercise

## ■ 1.1 – Your Role in Software Project Management

C A L T E C H
Project Management and Software Development
Exercise Guide
for <b>Boeing Satellite Systems</b>
California Institute of Technology Industrial Relations Center 380 South Hill Avenue

