Management Information Systems

Larisa Cherkasov, MBA

Martin V. Smith School of Business and Economics

CSU Channel Islands

Email: Larisa.Cherkasov@csuci.edu

Channel Islands
CALIFORNIA STATE UNIVERSITY

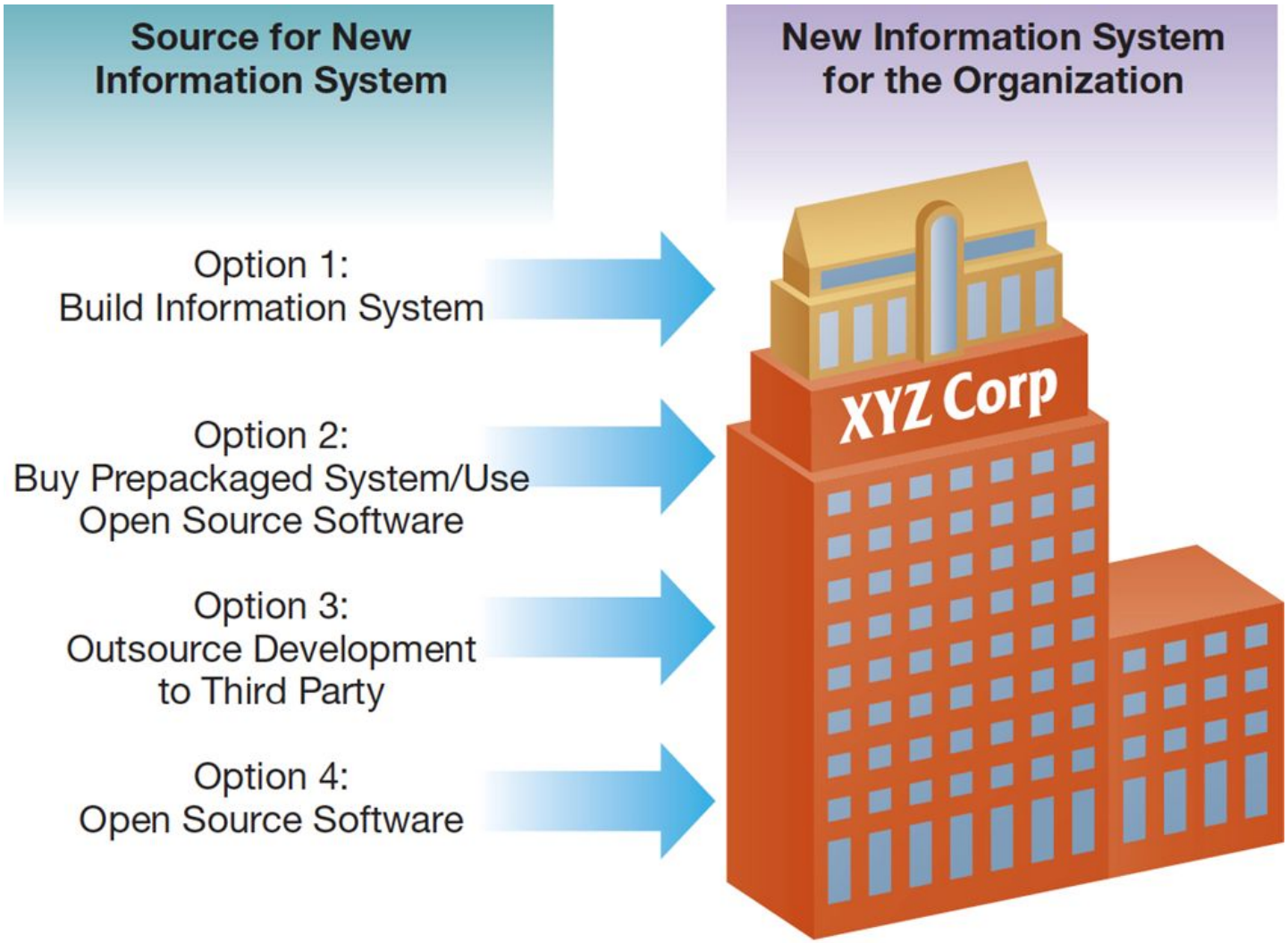
Developing and Acquiring Information Systems

Channel Islands
CALIFORNIA STATE UNIVERSITY

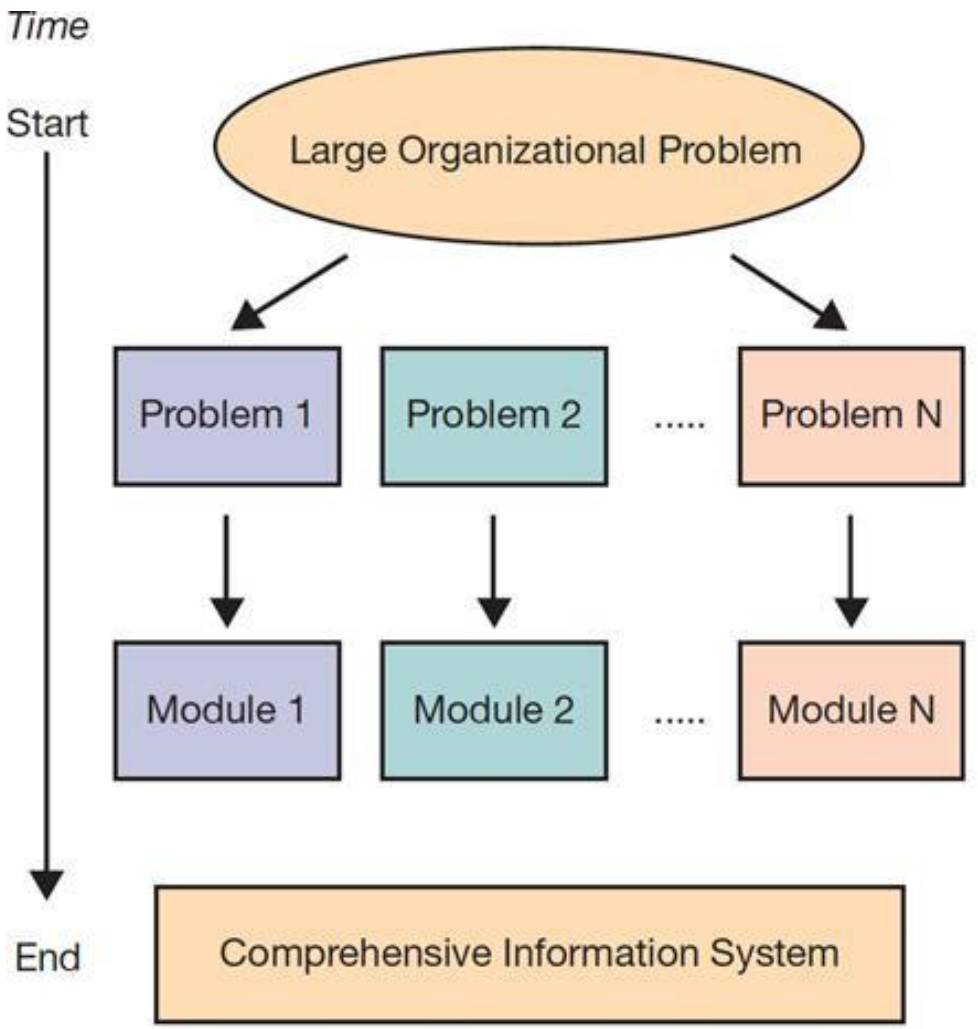
The Systems Development Process

- IS Development in Action
- The Role of Users in the Systems Development Process
- Systems Development Controls
- Steps in the Systems Development Process
  - Phase 1: Systems Planning and Selection
  - Phase 2: Systems Analysis
  - Phase 3: Systems Design
  - Phase 4: Systems Implementation and Operation
- Repeating the SDLC: Systems Maintenance
- Other Approaches to Designing and Building Systems

IS Development



IS Development



System Construction Process

Identify Large Organizational Problem to Solve

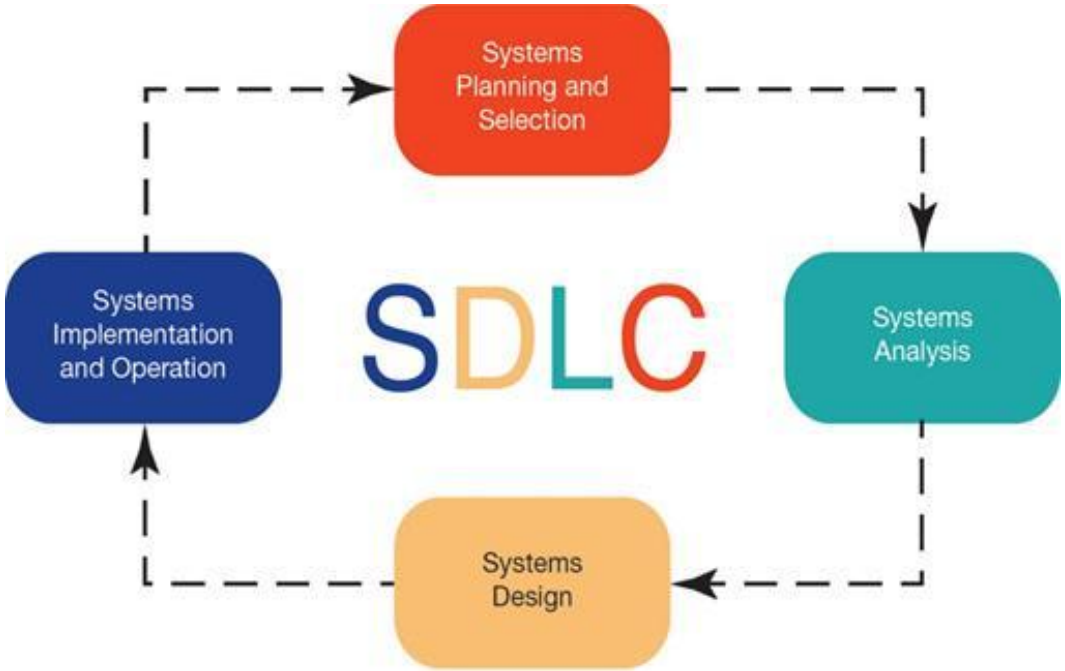
Decompose Large Problem into Many Small Problems

Translate Each Small Problem into a Software Module

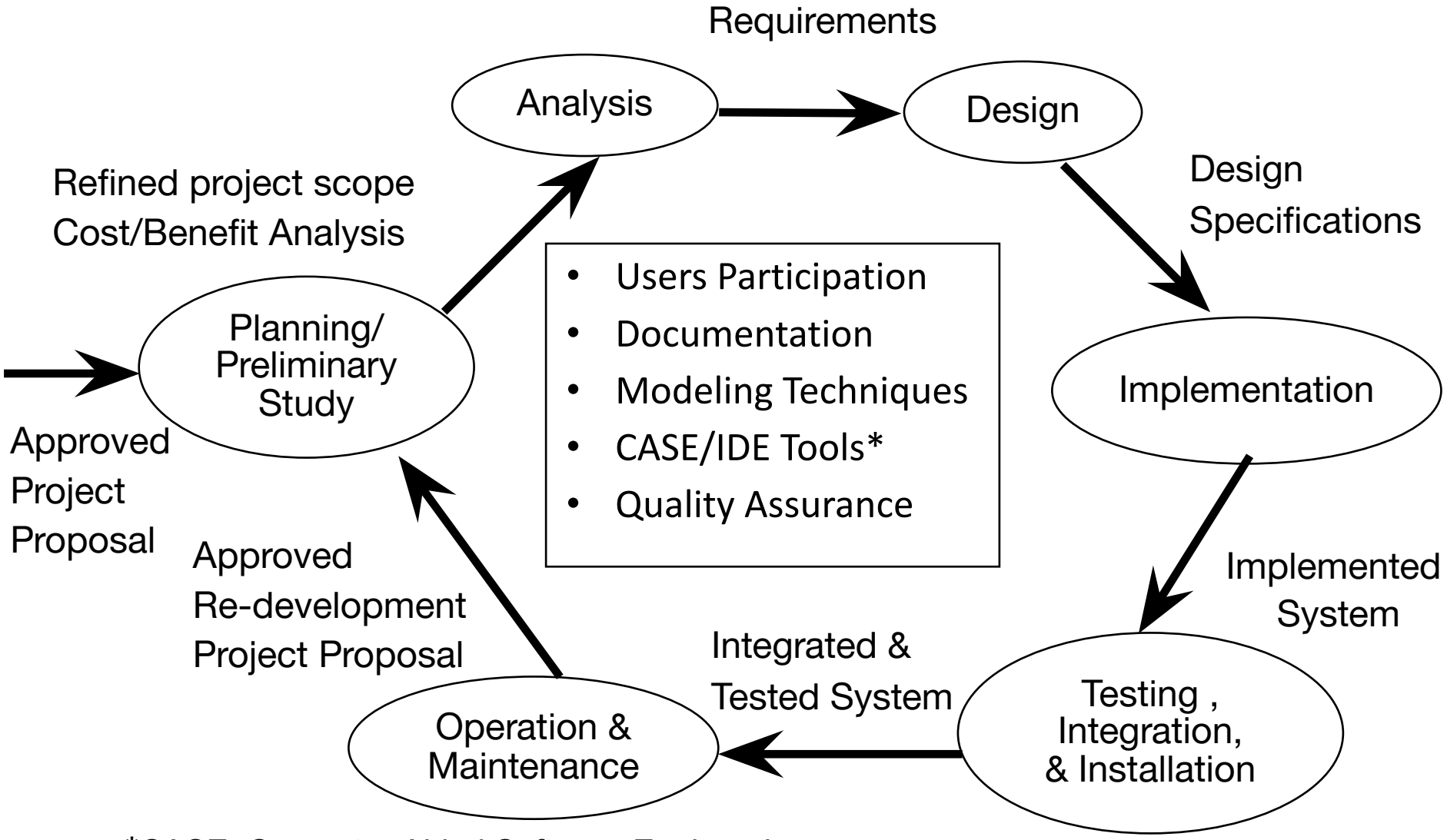
Piece Together the Separate Modules into an Overall Comprehensive Information System

Systems Development Process: Steps

1. Systems planning and selection
2. Systems analysis
3. Systems design
4. Systems implementation and operation

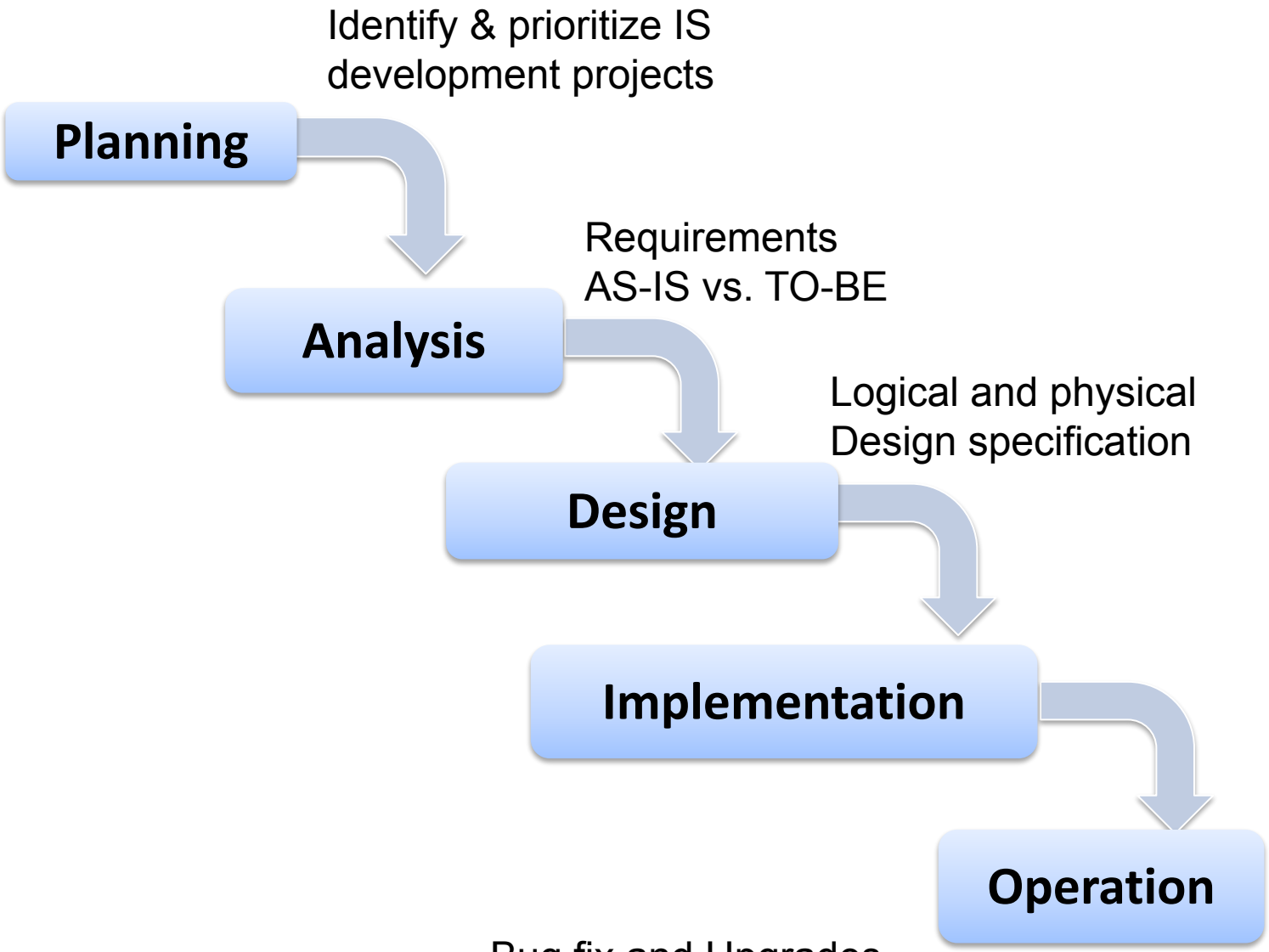


Systems Development Life Cycle (SDLC)



*CASE: Computer-Aided Software Engineering
IDE: Integrated Development Environment

SDLC Waterfall Model



SDLC: Deliverables & Documentations

Table 1-2 Products of SDLC Phases

Phase	*Products, Outputs, or Deliverables*
Planning	Priorities for systems and projects; an architecture for data, networks, and selection hardware, and IS management are the result of associated systems;
	Detailed steps, or work plan, for project;
	Specification of system scope and planning and high-level system requirements or features;
	Assignment of team members and other resources;
	System justification or business case
Analysis	Description of current system and where problems or opportunities are with a general recommendation on how to fix, enhance, or replace current system;
	Explanation of alternative systems and justification for chosen alternative
Design	Functional, detailed specifications of all system elements (data, processes, inputs, and outputs);
	Technical, detailed specifications of all system elements (programs, files, network, system software, etc.);
	Acquisition plan for new technology
Implementation	Code, documentation, training procedures, and support capabilities
Maintenance	New versions or releases of software with associated updates to documentation, training, and support

Phase 1: Systems Planning and Selection

- Resources are limited so projects must be limited
- Analyst gathers information and builds the case
- Multiple approaches to selecting projects
  - Formal IS planning process
  - Ad-hoc planning process
- The business case role
  - Business cases for different projects compared
  - Multiple selection criteria

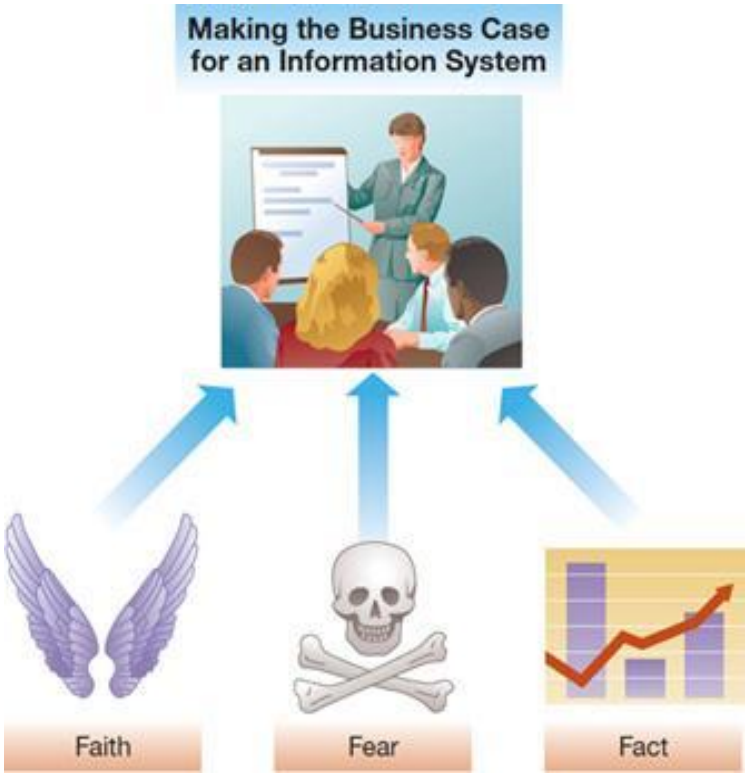
Business Case: Objectives

- **Business Case** is a complete justification for making or continuing to make an investment in a new or ongoing information system
- The business case sells an investment
  - Build a strong, integrated set of arguments
  - Show how an IS adds value to the organization
  - Lay out the costs and benefits
  - Used to make a “go” or “no-go” decision
  - May be used to justify continued funding

Making Business Case

- Process of identifying, quantifying, and presenting the value provided by a system

Type of Argument	Description
Faith	Arguments based on beliefs about organizational strategy, competitive advantage, industry forces, customer perceptions, market share, and so on
Fear	Arguments based on the notion that if the system is not implemented, the firm will lose out to the competition
Fact	Arguments based on data, quantitative analysis, and/or indisputable factors



Business Case: Costs and Benefits

- Identifying Costs
  - Tangible costs : Total cost of ownership (TCO)
    - Non-recurring costs
    - Recurring costs
  - Intangible costs
    - Loss of customers
- Identifying Benefits
  - Tangible benefits
    - Estimated sales gains
  - Intangible benefits
    - Improved customer service

Business Case: Cost-Benefit Analyses

- **Total Cost of Ownership** (TCO)
  - Total cost of *acquisition*

    AND
  - Costs of ongoing *use and maintenance* of a system
- Two Categories of Costs
  - Non-recurring costs
    - One-time costs that are not expected to continue after the system is implemented
    - Capital Expenditure
      - Acquisition
  - Recurring costs
    - Ongoing costs that occur throughout the life of the system
    - Operational Expenditure
      - Use and maintenance

Business Case: Cost-Benefit Analyses

		2018	2019	2020	2021	2022
Costs						
Non-recurring						
Hardware		$ 20,000				
Software		$ 7,500				
Networking		$ 4,500				
Infrastructure		$ 7,500				
Personnel		$100,000				
Recurring						
Hardware			$ 500	$ 1,000	$ 2,500	$ 15,000
Software			$ 500	$ 500	$ 1,000	$ 2,500
Networking			$ 250	$ 250	$ 500	$ 1,000
Service fees			$ 250	$ 250	$ 250	$ 500
Infrastructure				$ 250	$ 500	$ 1,500
Personnel			$ 60,000	$ 62,500	$ 70,000	$ 90,000
Total costs		$139,500	$ 61,500	$ 64,750	$ 74,750	$110,500
Benefits						
Increased sales		$ 20,000	$ 50,000	$ 80,000	$115,000	$175,000
Error reduction		$ 15,000	$ 15,000	$ 15,000	$ 15,000	$ 15,000
Cost reduction		$100,000	$100,000	$100,000	$100,000	$100,000
Total benefits		$135,000	$165,000	$195,000	$230,000	$290,000
Net costs/benefits		$ (4,500)	$103,500	$130,250	$155,250	$179,500

- Break-even analysis
  - At what point tangible benefits equal tangible costs
- Net-present-value analysis
  - Cash flow streams at the organization's discount rate

Business Case: Comparing Investments

- Weighted Multicriteria Analysis

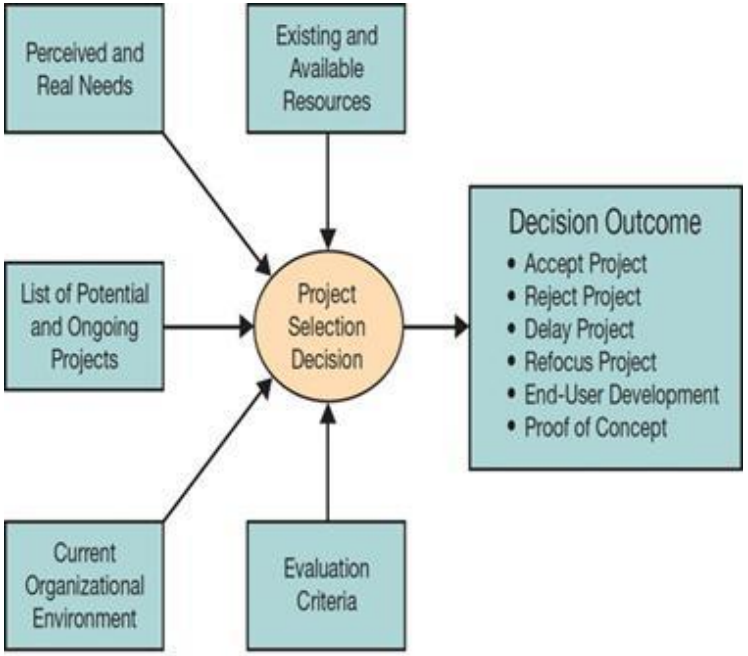
Criteria	Weight	Alternative A		Alternative B		Alternative C	
		Rating	Score	Rating	Score	Rating	Score
Requirements							
Web-based Interface	18	5	90	5	90	5	90
Security capabilities	18	1	18	5	90	5	90
BI capabilities	14	1	14	5	70	5	70
	50		122		250		250
Constraints							
Software Costs	15	4	60	5	75	3	45
Hardware Costs	15	4	60	4	60	3	45
Operating Costs	15	5	75	1	15	5	75
Ease of Training	5	5	25	3	15	3	15
	50		220		165		180
Total	100		342		415		430

Business Case: Presenting

- Know the Audience
  - Know who you are presenting to, what their background is, and what they care about
- Convert Benefits to Monetary Terms
  - Show benefits as $ per time period, often annual
- Devise Proxy Variables
  - A proxy variable is a variable that is relevant to the audience if your normal metrics aren't
- Measure What Is Important to Management
  - Know management "hot-button" issues
  - Describe how the system impacts them

Business Case: Stakeholders

Stakeholder	Focus and Project Characteristics
Management	Greater strategic focus; largest project sizes; longest project durations
Steering Committee	Cross-functional focus; greater organizational change; formal cost-benefit analysis; larger and riskier projects
User department	Narrow, no-strategic focus; faster development
IS Executive	Focus on integration with existing systems; fewer development delays; less concern with cost-benefit analysis



Phase 2: Systems Analysis

- Collecting Requirements is the process of gathering and organizing information from users by the following means:

  - Interviews
  - Questionnaires
  - Observations
  - Document Analysis
  - Joint Application Design (JAD)
    - A group meeting-based process for requirements collection

Phase 2: Systems Analysis

- **Business Requirement Document** (BRD) describes the high-level business needs
  - Answers the question what the business wants to do

1) Identify Key Stakeholders
   - Who has the final say on what will be included in the project's scope
   - Who will use the solution, product, or service
2) Capture Stakeholder Requirements
   - What do they want and expect from this project?

https://www.mindtools.com/pages/article/newPPM_77.htm

Phase 2: Systems Analysis

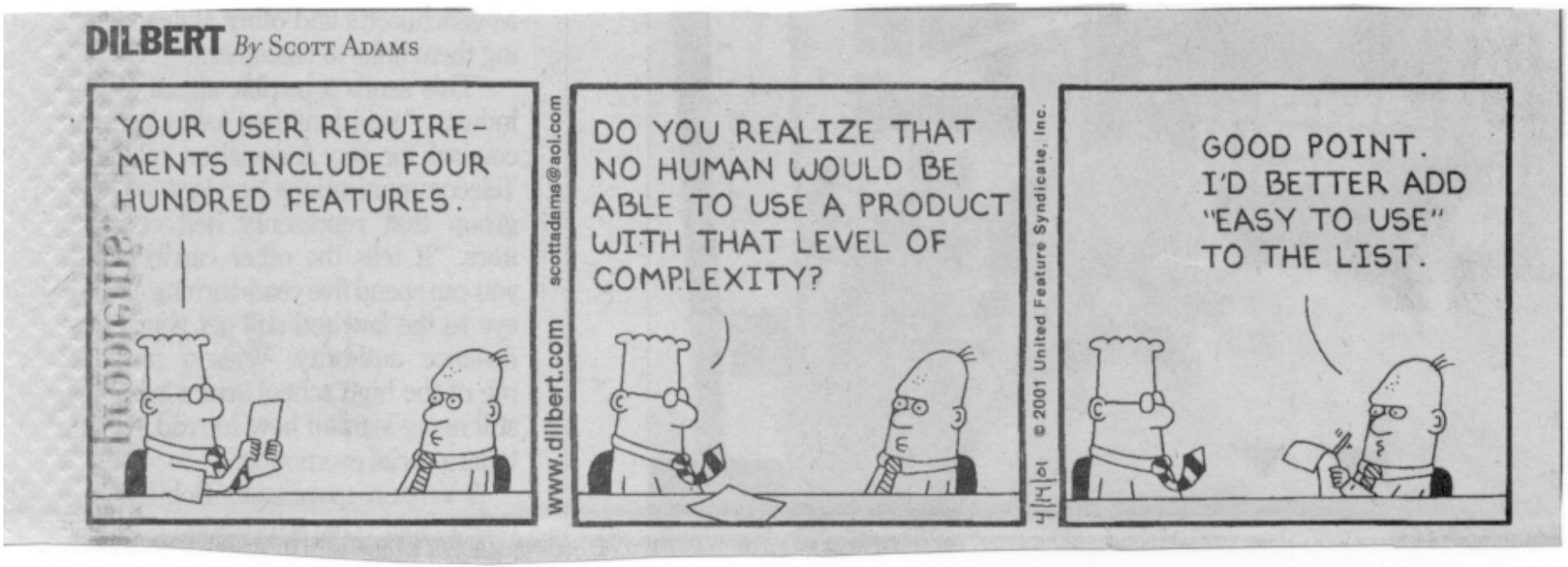
3) Categorize, Interpret and Record Requirements

- Define requirements precisely – Ensure that the requirements are:
  - Not ambiguous or vague
  - Clearly worded.
  - Sufficiently detailed so that everything is known
  - Related to the business needs.
  - Listed in sufficient detail to create a working system
- Prioritize requirements
- Analyze the impact of change
- Resolve conflicting issues
- Analyze feasibility

4) Sign Off

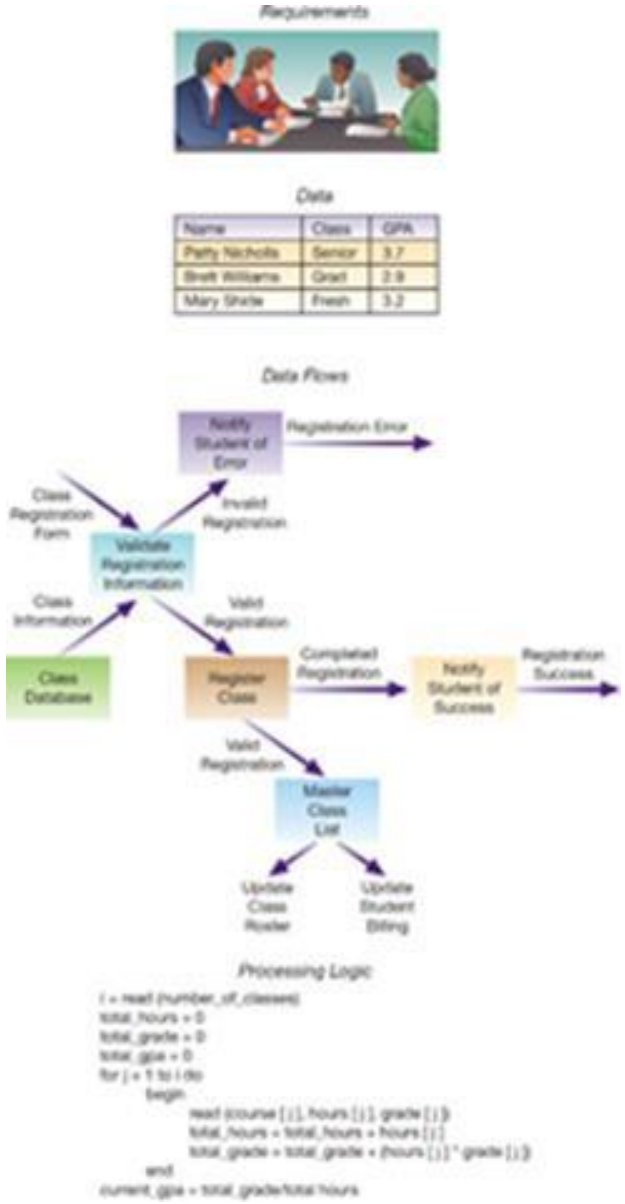
- Get the signed agreement of key stakeholder

Phase 2: Systems Analysis



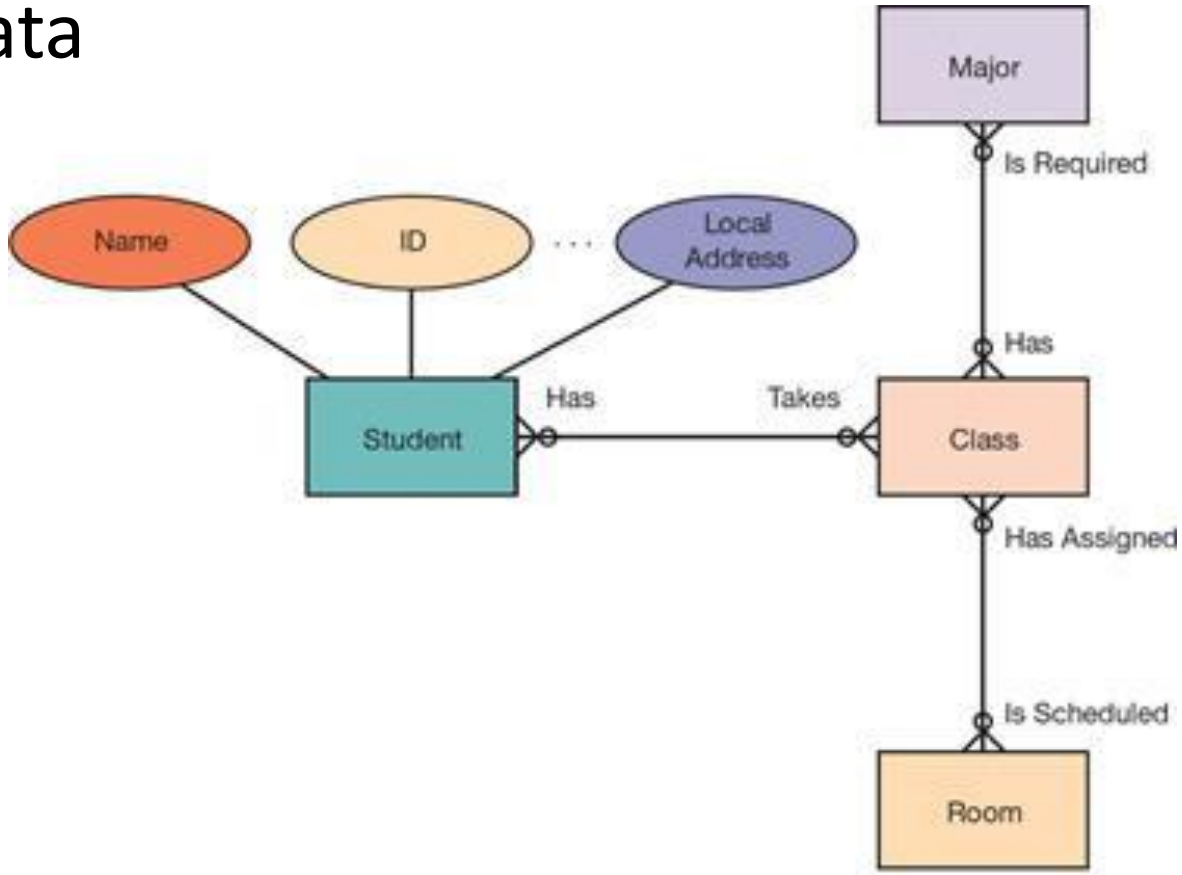
Phase 2: Systems Analysis

- Modeling Processes and Logic
  - Key elements in the system development
    - Requirements
    - Data
    - Data flows
    - Processing logic



Phase 2: Systems Analysis

- Modeling Data



Phase 3: Systems Design

- The system is designed based on the chosen design during Phase 2:
  - **Functional Requirement Document (FRD)** outlines the functions required to fulfill the business need
    - Functions that the system must perform in order to fulfill the Business Requirements.
  - Technical Specifications
  - Processing and Logic
  - Modeled using one of many techniques
    - Models converted into code in Phase 4
  - Database and files
  - Human-computer interface
  - Point of contact between the user and the system
  - Data entry and management forms

Phase 4: Systems Implementation and Operation

- Convert design into a working system
  - Software programming and software testing
  - System conversion, documentation, training, and support
    - User and reference guides
    - User training manuals and tutorials
    - Installation procedures and troubleshooting suggestions

Phase 4: Systems Implementation and Operation

- Brook's Law:
  - Adding developers to a late project will make it later.

Phase 4: Types of Software Testing

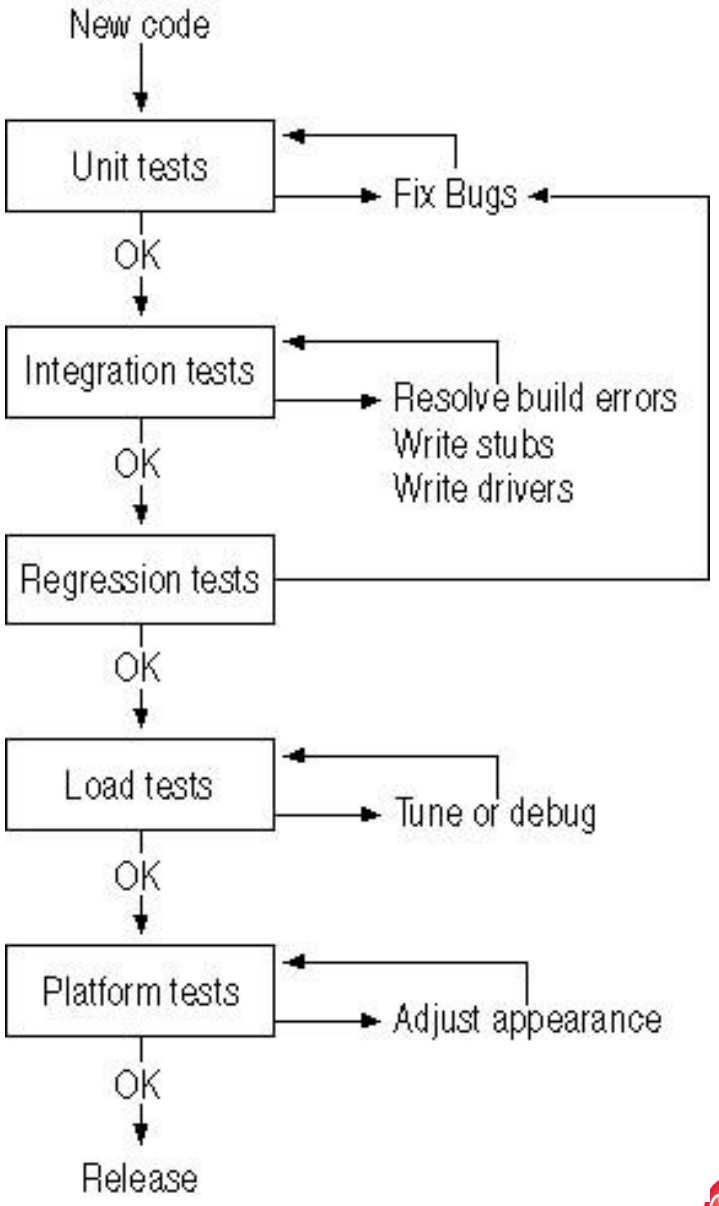
Testing Type	Focus	Performed by
Developmental	Testing the correctness of individual modules and the integration of multiple modules	Programmer
Alpha	Testing of overall system to see whether it meets design requirements	Software tester
Beta	Testing of the capabilities of the system in the user environment with actual data	Actual system users

Phase 4: Types of Software Testing

Testing Types	***Objectives***
Unit test	Each independent piece of code works correctly
Integration test	All units work together without errors
Regression test	Newly added features do not introduce errors to other features that are already working
Load test (also called stress test)	The product continues to work under extreme usage
Platform test	The product works on all of the target hardware and software platforms

Phase 4: Software Testing

- Test plan objectives
  - Is thoroughly tested
  - Meets requirements
  - Does not contain defects
- Test plan covers
  - Tools
  - Who
  - Schedule
- Test result analysis
  - What is being tested?
- Test cases
  - Automated testing
  - Reproducible
  - Measurable



Phase 4: Software Testing

- **Regression Test:** Process of validating modified parts of the software and ensuring that no new errors are introduced into previously tested code.
  - Unit and integration tests form the basis of regression testing.
  - As each test is written and passed, it gets checked into the test library for a regularly scheduled testing run.
  - If a new component or a change to an existing component breaks one of the existing unit or integration tests, the error is called a **regression**.

SDLC Waterfall Model: The Advantages

- Allows to maintain a more robust scope and design structure due to all the upfront planning and documentation stages.
  - Adapts to shifting teams
- Forces the project to be extraordinarily disciplined in its design and structure.
  - Include detailed procedures to manage every aspect of the project, from design and development to testing and implementation.
- Allows for early design changes
  - Alterations early in the life cycle can be made immediately and with minimal effort, since no coding or implementation has actually taken place up to that point.
- Suited for milestone-focused development
  - Relatively simple to develop a timeline for the process and assign particular milestones for each stage and even completion

SDLC Waterfall Model: The Disadvantages

- Nonadaptive Design Constraints
  - Inherent lack of adaptability across all stages of the development life cycle
- Ignores Mid-Process User/Client Feedback:
  - User or client feedback that is provided late into the development cycle can often be too little, too late.
- Delayed Testing Period
  - Most bugs or even design issues won't be discovered until very late into the process

System Development Frameworks

Agile software development (Agile)	
Pros	Minimizes feature creep by developing in short intervals resulting in miniature software projects and releasing the product in mini-increments.
Cons	Short iteration may add too little functionality, leading to significant delays in final iterations. Since Agile emphasizes real-time communication (preferably face-to-face), using it is problematic for large multi-team distributed system development. Agile methods produce very little written documentation and require a significant amount of post-project documentation.
Scrum	
Pros	Improved productivity in teams previously paralyzed by heavy “process”, ability to prioritize work, use of backlog for completing items in a series of short iterations or sprints, daily measured progress and communications.
Cons	Reliance on facilitation by a master who may lack the political skills to remove impediments and deliver the sprint goal. Due to relying on self-organizing teams and rejecting traditional centralized "process control", internal power struggles can paralyze a team.
Lean software development (LD)	
Pros	Creates minimalist solutions (i.e., needs determine technology) and delivers less functionality earlier; per the policy that 80% today is better than 100% tomorrow.
Cons	Product may lose its competitive edge because of insufficient core functionality and may exhibit poor overall quality.
Rapid application development (RAD)	
Pros	Promotes strong collaborative atmosphere and dynamic gathering of requirements. Business owner actively participates in prototyping, writing test cases and performing unit testing.
Cons	Dependence on strong cohesive teams and individual commitment to the project. Decision making relies on the feature functionality team and a communal decision-making process with lesser degree of centralized PM and engineering authority.

SD Frameworks: Agile Manifesto

http://agilemanifesto.org/

SD Frameworks : Agile Manifesto

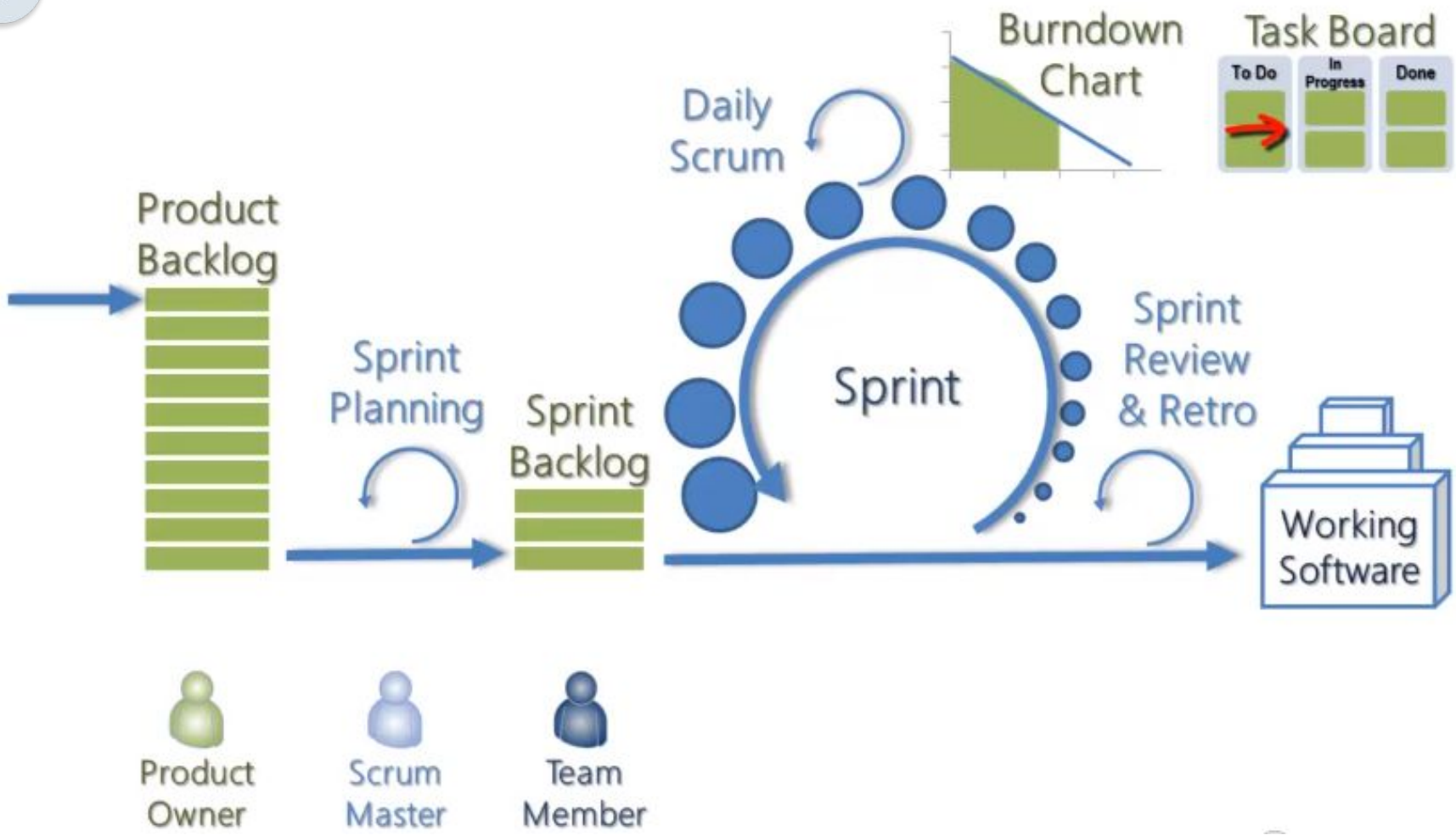
Individuals and interactions	over	Process and tools
Working software	over	Comprehensive documentation
Customer collaboration	over	Contract negotiation
Responding to change	over	Following a plan

http://agilemanifesto.org/

SD Frameworks : SCRUM Definition

- **SCRUM** is an agile process that allows to focus on delivering the highest business value in the shortest time.
  - Rapidly and repeatedly inspect actual working software.
  - The business sets the priorities.
  - Teams self-organize to determine the best way to deliver the highest priority features.
  - Every two weeks to a month anyone can see real working software and decide to release it as is or continue to enhance it for another sprint.

SD Frameworks: SCRUM Process



SCRUM Process

SD Frameworks : SCRUM Framework

Roles

- Product owner
- ScrumMaster

Ceremonies

- Sprint planning
- Sprint review
- Sprint retrospective

Artifacts

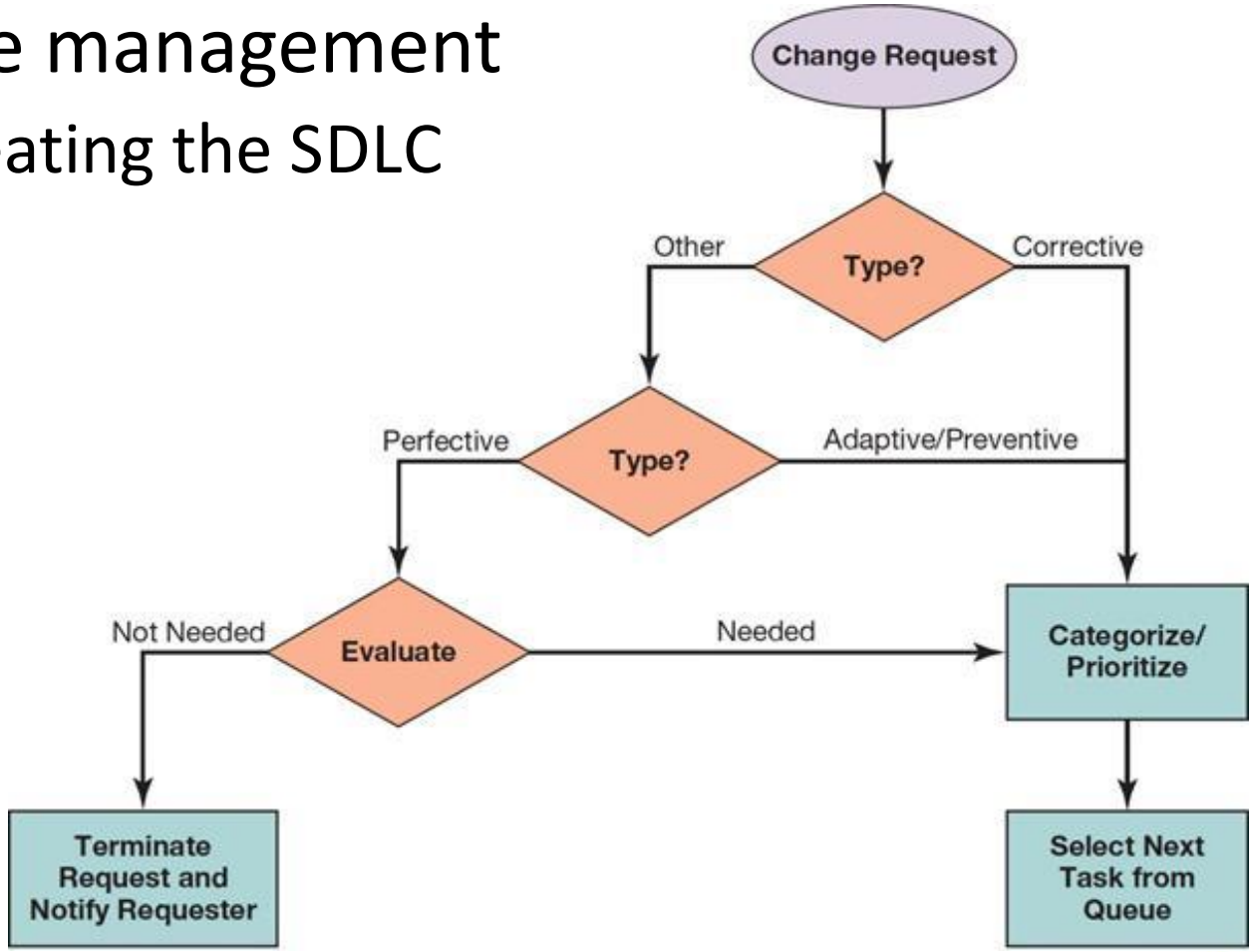
- Product backlog
- Sprint backlog

SD Frameworks: SCRUM Characteristics

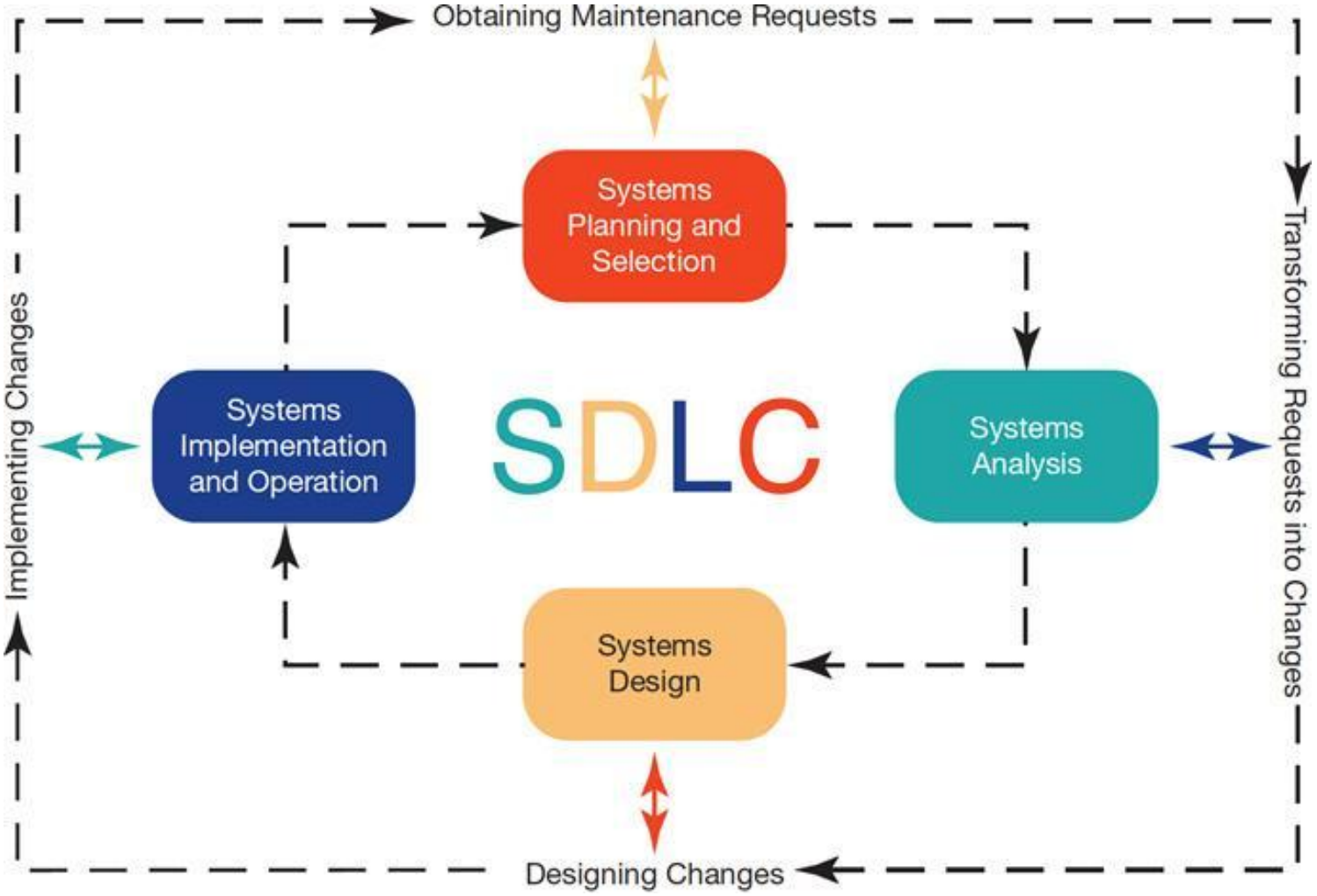
- Teams of three to nine members
  - The team is given clear goals
  - The team organizes itself around the work
  - The team regularly delivers the most valuable features
  - The team receives feedback from people outside it
  - The team reflects on its way of working in order to improve
  - The team and management honestly communicate about progress and risks
- Sprints
  - Analogous to Extreme Programming iterations
  - Typical duration is 2–4 weeks or a calendar month at most
  - A constant duration leads to a better rhythm
  - Product is designed, coded, and tested during the sprint

Systems Maintenance

- Regular changes, upgrades, and maintenance
- Change management
  - Repeating the SDLC



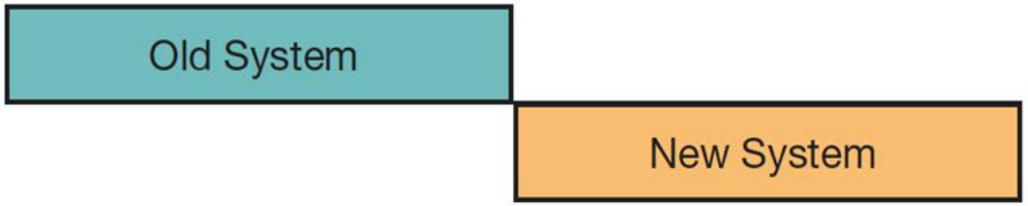
Systems Maintenance



Conversion Strategies

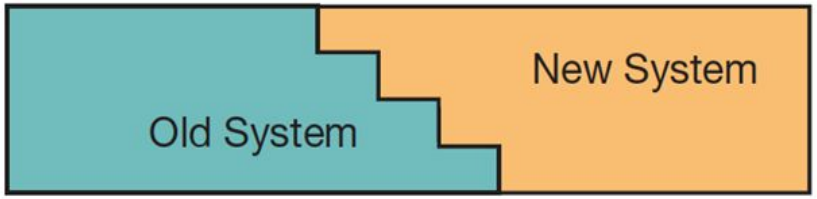
(a) Parallel

Old and new systems are used at the same time.



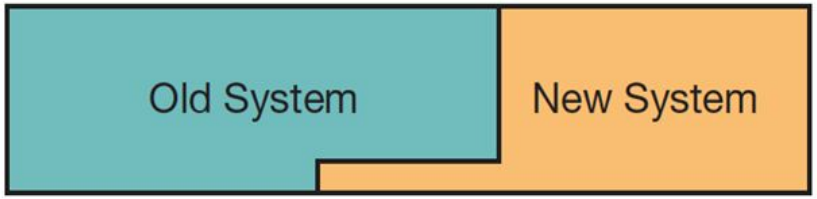
(b) Direct

Old system is discontinued on one day, and the new is used on the next.



(c) Phased

Parts of the new system are implemented over time.



(d) Pilot (single location)

Entire system is used in one location.

Description

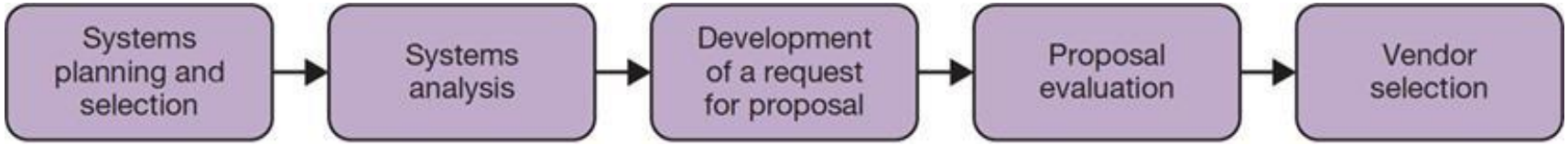
Channel Islands
CALIFORNIA STATE UNIVERSITY

External Acquisition: Reasons

- Possible situations:
  - Situation 1
    - Limited IS staff
  - Situation 2
    - IS staff has limited skill set
  - Situation 3
    - IS staff is overworked
  - Situation 4
    - Problems with performance of IS staff
- Options:
  - External acquisition of a prepackaged system
  - Outsourcing systems development

External Acquisition: Request for Proposal

- A summary of existing systems and applications
- Requirements for system performance and features
- Reliability, backup, and service requirements
- The criteria that will be used to evaluate proposals
- Timetable and budget constraints



External Acquisition: Evaluation Criteria

Hardware Criteria	Software Criteria	Other Criteria
Brand/manufacturer	Business alignment	Installation/Training
Speed/Storage	Required/desired features	Vendor characteristics (years in business, flexibility, reputation, etc.)
Reliability	Reliability	Price
Scalability for growth	Operation integration	
Ease of installation	Scalability for growth	
Ease of integration	Support model	
Warranty	Usability	

External Acquisition: Vendor Selection

- Typically multiple feasible solutions
- Prioritize or rank competing proposals
- Weighted scoring system works well for this
- Other approaches include:
  - Simple checklists
  - Subjective processes
- Once vendor is selected, external acquisition is complete

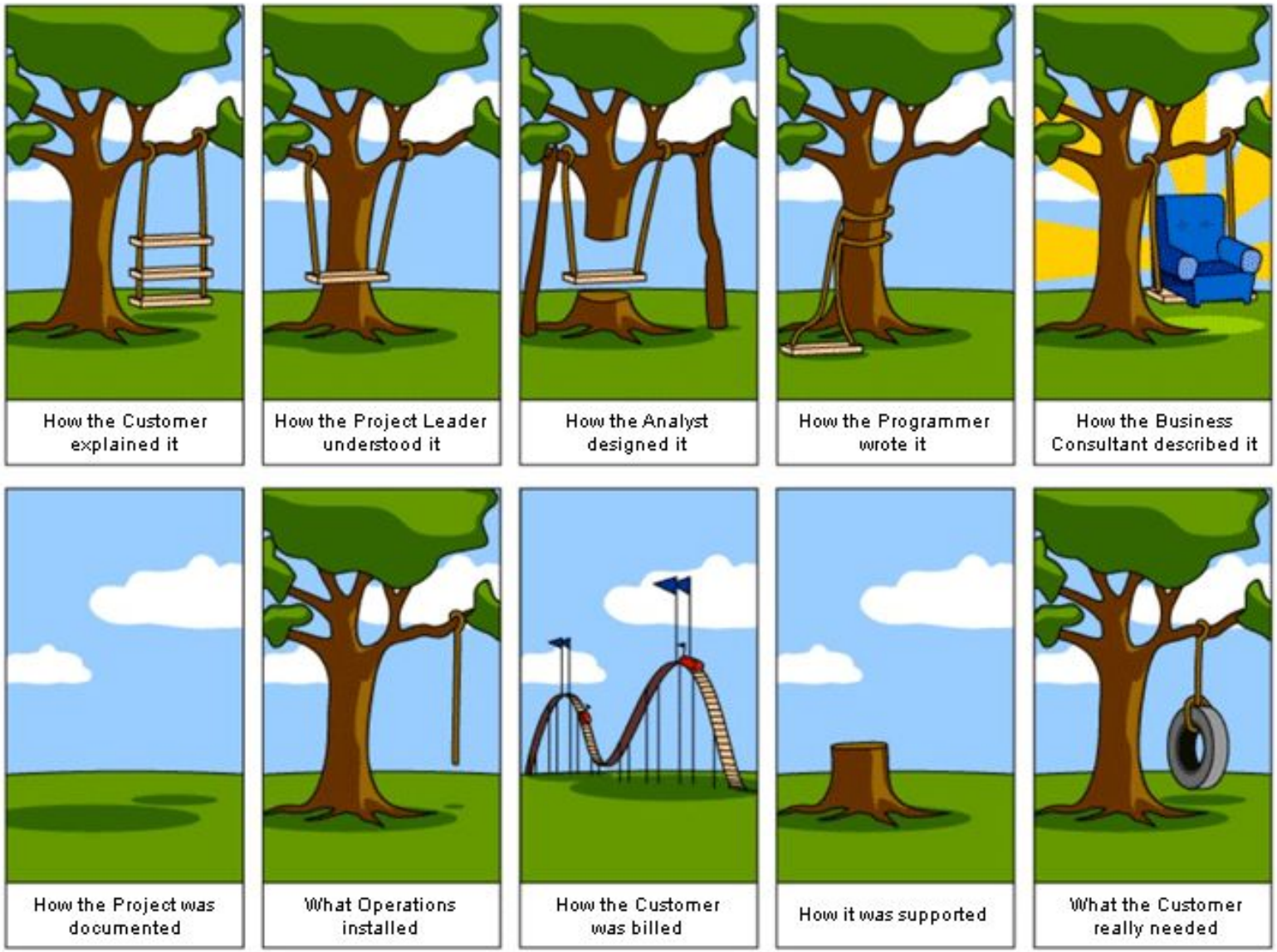
Outsourcing: Reasons

Outsourcing Reasons	Description
Cost and Quality Concerns	Vendors have economies of scale, and can develop better systems at a lower cost
Problems in IS Performance	Outsourced vendors are more reliable and consistent
Supplier Pressure	Aggressive sales forces
Simplifying, Downsizing, and Reengineering	Companies retreating to core competencies, outsourcing functions not core to value creation
Financial Factors	An arm's-length relationship with vendors can create more efficient use, IT assets can be liquidated
Organizational Culture	Internal politics may block IT from moving forward
Internal Irritants	If the IS staff and users are not interacting well together, removing that source of tension can be a relief

Outsourcing: Managing Relationship

- Outsourced relationships take continuous management
- Realistic, tangible measures of performance should be developed and tracked
- Multiple levels of interaction based on the type of interaction
  - Operational and tactical
  - Policy and relationship

Reasons for Project Failures



Reasons for Project Failures

- Primary reasons for project failure include
  - Unclear or missing business requirements
  - Skipping SDLC phases
  - Failure to manage project scope
    - **Scope creep** – occurs when the scope increases
    - **Feature creep** – occurs when extra features are added
  - Failure to manage project plan
  - Changing technology

Reasons for Project Failures

Why Do Technology Projects Fail?

- Unrealistic or unclear project goals
- Poor project leadership and weak executive commitment
- Inaccurate estimates of needed resources
- Badly defined system requirements and allowing "feature creep" during development
- Poor reporting of the project's status
- Poor communication among customers, developers, and users
- Use of immature technology
- Unmanaged risks
- Inability to handle the project's complexity
- Sloppy development and testing practices
- Poor project management
- Stakeholder politics
- Commercial pressures

Sources:
Joseph Valacich, Christoph Schneider, *Information Systems Today: Managing in the Digital World*, 8th Edition
John Gallaugher, Information Systems: A Manager's Guide to Harnessing Technology, v. 7.0
Minder Chen,Ph.D., Management Information Systems Lectures
Mitri, M., Cole, C., & Atkins, L. (2017). Teaching Case: A Systems Analysis Role-Play Exercise and Assignment. *Journal of Information Systems Education,* 28(1), 1-10.
http://jise.org/Volume28/n1/JISEv28n1p1.pdf