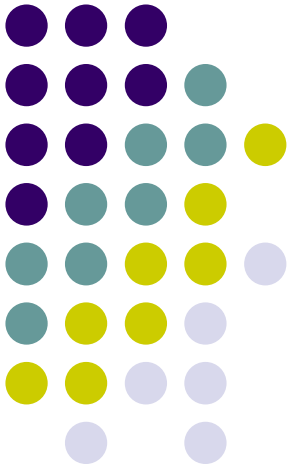
IT Project Management [INSY 714 / HUMA 1001]

GUC - Spring 2024 – Lecture 2

System Development Lifecycle



Dr. Ayman Al-Serafi & Dr. Noha Amer

TAs:

INSY/BINF: Farida Kamel, Ahmed Maged, Farida Waleed

HUMA: Farida Elhussainy, Mariam Elemary, Salma Ahmed, Seif Elabsy

Outline

1. Systems Development Project
2. Traditional Systems Development Approaches
3. Alternative Systems Development Approaches
4. Conclusion

Outline

1. **Systems Development Project**
2. Traditional Systems Development Approaches
3. Alternative Systems Development Approaches
4. Conclusion

IT / IS / SD Projects

- **IT project** refers to projects involving hardware, software, and networks
- **IS project** refers to projects involving applications that serve the informational needs of IS users (business applications)
- **Systems development** refers to projects producing new IS that did not exist before

German University in Cairo
الجامعة الألمانية بالقاهرة

Success…

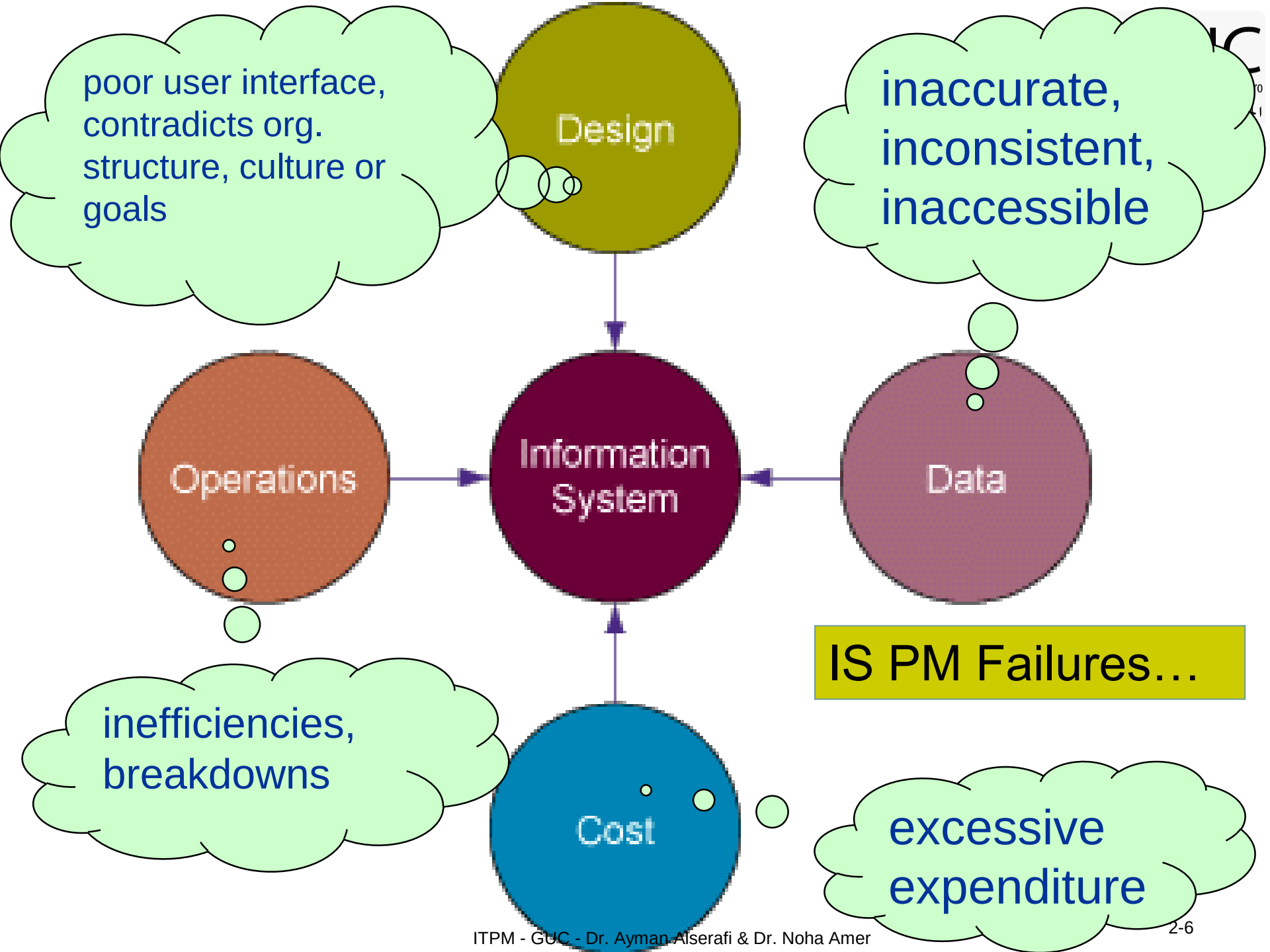
a project that delivers to the customer everything specified, to the quality agreed, on time and within costs

Challenged…

the project was completed and became operational but cost more, overran on time and delivered less functionality or bad quality

Impaired…

cancelled during the development stage


poor user interface, contradicts org. structure, culture or goals
Design
inaccurate, inconsistent, inaccessible
Operations
Information System
Data
IS PM Failures…
inefficiencies, breakdowns
Cost
excessive expenditure

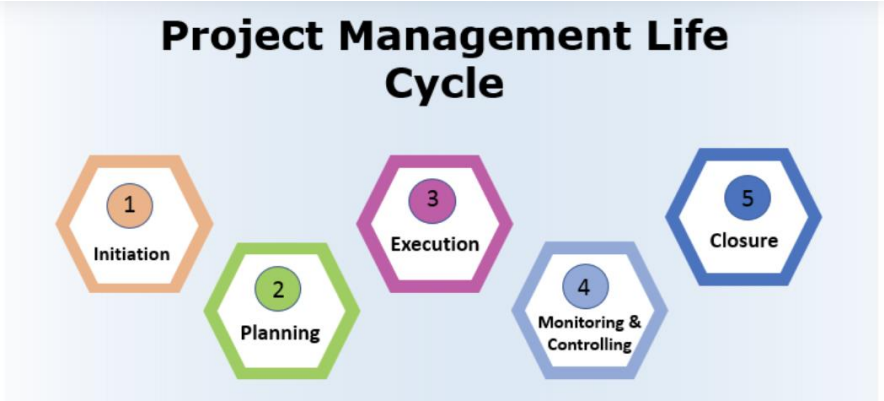
Outline

1. Systems Development Project
2. **Traditional Systems Development Approaches**
3. Alternative Systems Development Approaches
4. Conclusion

The Project Life Cycle

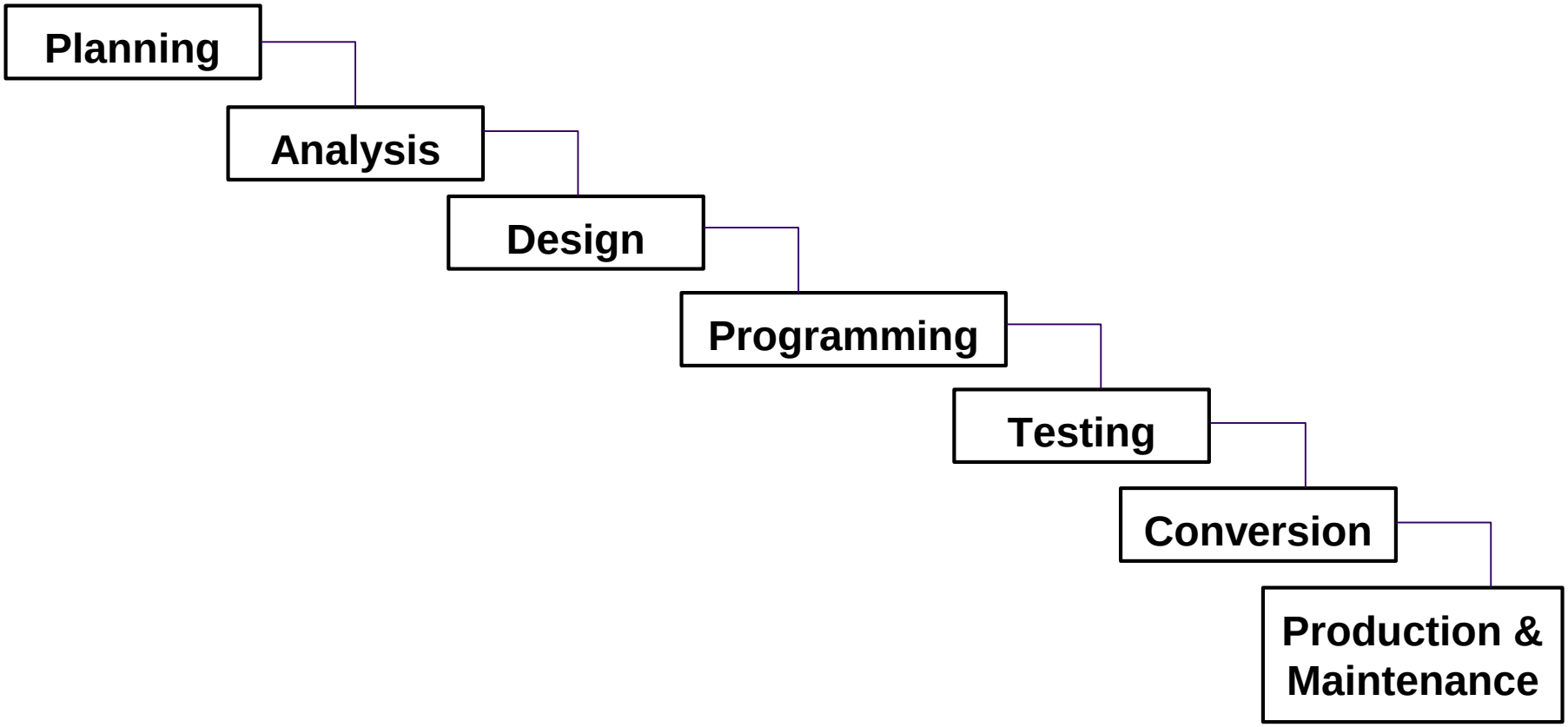
- A project life cycle is a collection of project phases

- Project phases vary by project or industry, but some general phases include
  - concept
  - development
  - implementation
  - support



The System Development Life Cycle (SDLC)

Activities in a **regular process of ordered steps** that go into **producing an information system solution** to an **organizational problem** or **opportunity**



Difference between Project Management Lifecycle (PMLC) & Software Development Lifecycle (SDLC)

- **PMLC** is concerned with the management of the project as a whole, including resources, schedules, budgets, risks, and stakeholders.
  - PMLC focuses on managing the project to ensure it's completed on time, within budget, and meets stakeholders' expectations.
- **SDLC** is primarily concerned with the development of a system or software product.
  - SDLC focuses on the technical aspects of product development

- Which one is broader in scope?

- **System Planning**
  - Project initiation
    - Prepare system request
    - Perform preliminary feasibility analysis
  - Set up the project
    - Project plan, including work plan and staffing plan

•Overview of Systems Development

Systems analysis

- **Analysis of business problem**
  - **Defining** the **problem** and **identifying causes**
  - **Specifying solutions**
    - Written systems proposal **report** describes costs and benefits of each alternative solution
  - Identifying **information requirements** to be met
    - **Who** needs **what** information **where**, **when**, and **how**

- Includes **feasibility study**
  - Is the solution a **good investment**?
  - Is the required **technology** skill **available**?

Systems design

- Describe **system specifications** that will deliver functions identified during systems analysis
- Should address all **managerial**, **organizational**, and **technological** components of system solution

- **Role of end users**
  - **User information requirements □ drive system-building**
  - Users must have **sufficient control over design process** to ensure that system **reflects their business priorities** and **information** needs

•Overview of Systems Development

Design Specifications

OUTPUT
Content
Timing

INPUT
Origins
Flow
Data entry

USER INTERFACE
Simplicity
Efficiency
Logic
Feedback
Errors

DATABASE DESIGN
Logical data model
Volume and speed requirements
File organization and design
Record specifications

PROCESSING
Computations
Program modules
Required reports
Timing of outputs

MANUAL PROCEDURES
What activities
Who performs them
When
How
Where

CONTROLS
Input controls (characters, limit, reasonableness)
Processing controls (consistency, record counts)
Output controls (totals, samples of output)
Procedural controls (passwords, special forms)

SECURITY
Access controls
Catastrophe plans
Audit trails

DOCUMENTATION
Operations documentation
Systems documents
User documentation

CONVERSION
Transfer files
Initiate new procedures
Select testing method
Cut over to new system

TRAINING
Select training techniques
Develop training modules
Identify training facilities

ORGANIZATIONAL CHANGES
Task redesign
Job redesign
Process design
Organization structure design
Reporting relationships

German University in Cairo
الجامعة الألمانية بالقاهرة

Programming:

- System **specifications from design stage** are translated into **software program code**
- **Software** may be **purchased**, **leased (subscription)**, or **outsourced**

German University in Cairo
الجامعة الألمانية بالقاهرة

Testing

- To **ensure** system produces **right results**
- **Test plan:** All preparations for **series of tests**
  - **Unit testing**: Tests **each program** in system **separately**
  - **System testing**: Tests **functioning** of **system as a whole**
- **(user-) Acceptance testing (UAT):** Makes sure system **is ready** to be used in production setting

German University in Cairo
الجامعة الألمانية بالقاهرة

Conversion

- Process of **changing from old system** to **new system**
- Four main strategies:
    1. **Parallel strategy**
    2. **Direct cutover**
    3. **Pilot study**
    4. **Phased approach**
- Requires **end-user training**
- **Finalization of detailed documentation** showing **how system works from technical and end-user standpoint**

German University in Cairo
الجامعة الألمانية بالقاهرة

Production and maintenance

- System **reviewed to determine** if any **revisions** **needed**
- May prepare formal **post-implementation audit** document

German University in Cairo
الجامعة الألمانية بالقاهرة

Maintenance

- **Changes** in **hardware**, **software**, **documentation**, or **procedures** to a **production system**
  - to correct **errors**,
  - meet **new requirements**,
  - or **improve processing** efficiency

Facts

- **20% debugging, emergency work**
- **20% changes to hardware, software, data, reporting**
- **60 % of maintenance work:**
  - **User enhancements; Improving** documentation; **Recoding system components** for greater processing efficiency

Summary of Systems Development Activities

CORE ACTIVITY	DESCRIPTION
Systems analysis	• Identify problem(s) • Specify solutions • Establish information requirements
Systems design	• Create design specifications
Programming	• Translate design specifications into code
Testing	• Unit test • Systems test • Acceptance test
Conversion	• Plan conversion • Prepare documentation • Train users and technical staff
Production and maintenance	• Operate the system • Evaluate the system • Modify the system

Outline

1. Systems Development Project
2. Traditional Systems Development Approaches
3. **Alternative Systems Development Approaches**
4. Conclusion

Alternative Systems-Building Approaches

1- Traditional systems lifecycle

2- Agile / Prototyping

3- End-user development

4- Application software packages & outsourcing

•Alternative Systems-Building Approaches

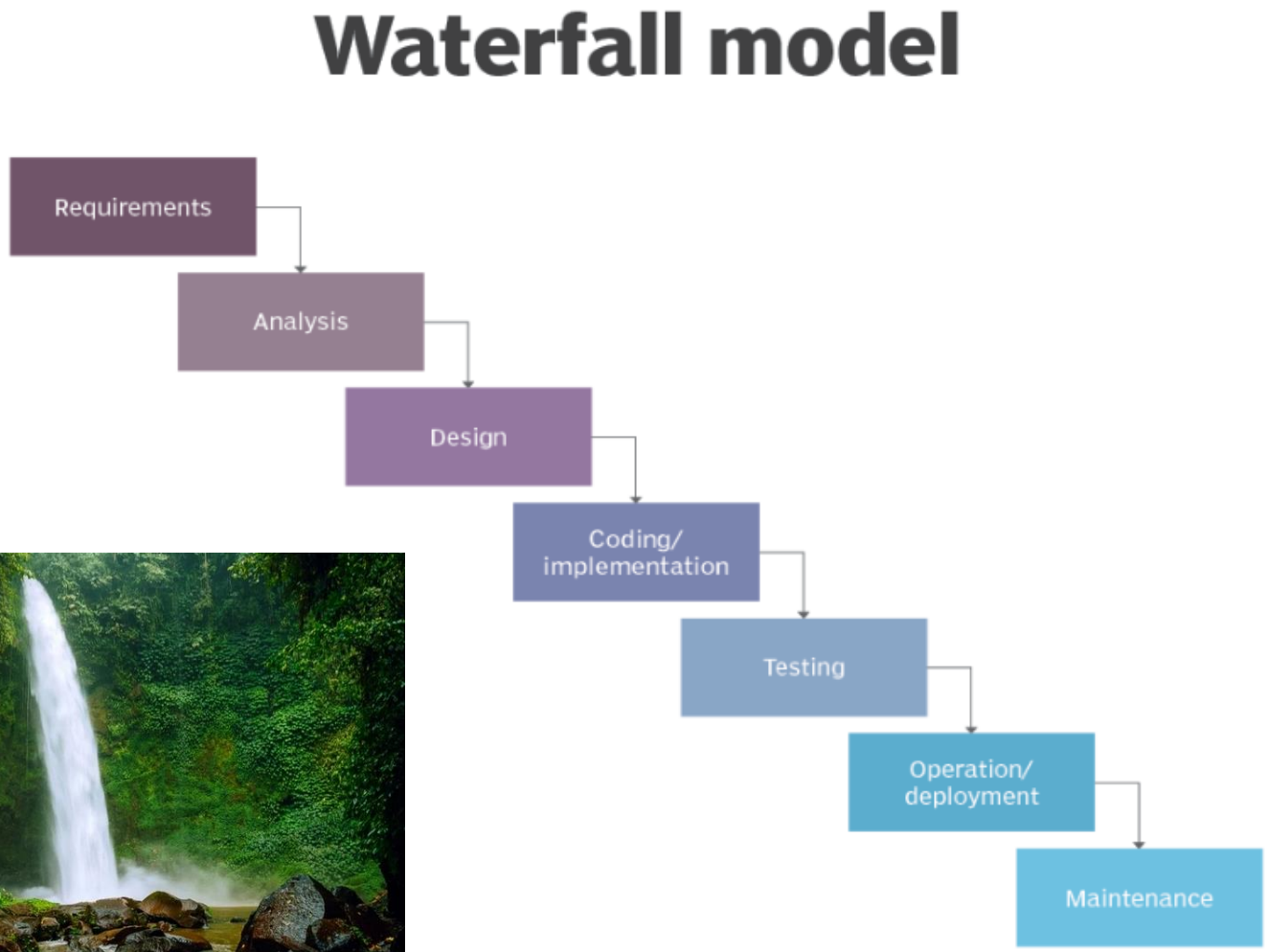
1- Traditional systems lifecycle:

- **Oldest method** for building information systems
- **Phased approach** - divides development into **formal stages**
- **Maintains formal division of labor** between
  - **End users** and
  - **Information systems specialists**
- Emphasizes **formal specifications** and **paperwork**
- **Still used** for building large complex systems that have known requirements

Disadvantages

- Can be **costly**, **time-consuming**, and **inflexible**

Waterfall Life Cycle Models



Waterfall Methodology Assessment

Strengths

- System requirements identified long before construction begins
- Requirements are “frozen” as project proceeds – no moving targets allowed

Weaknesses

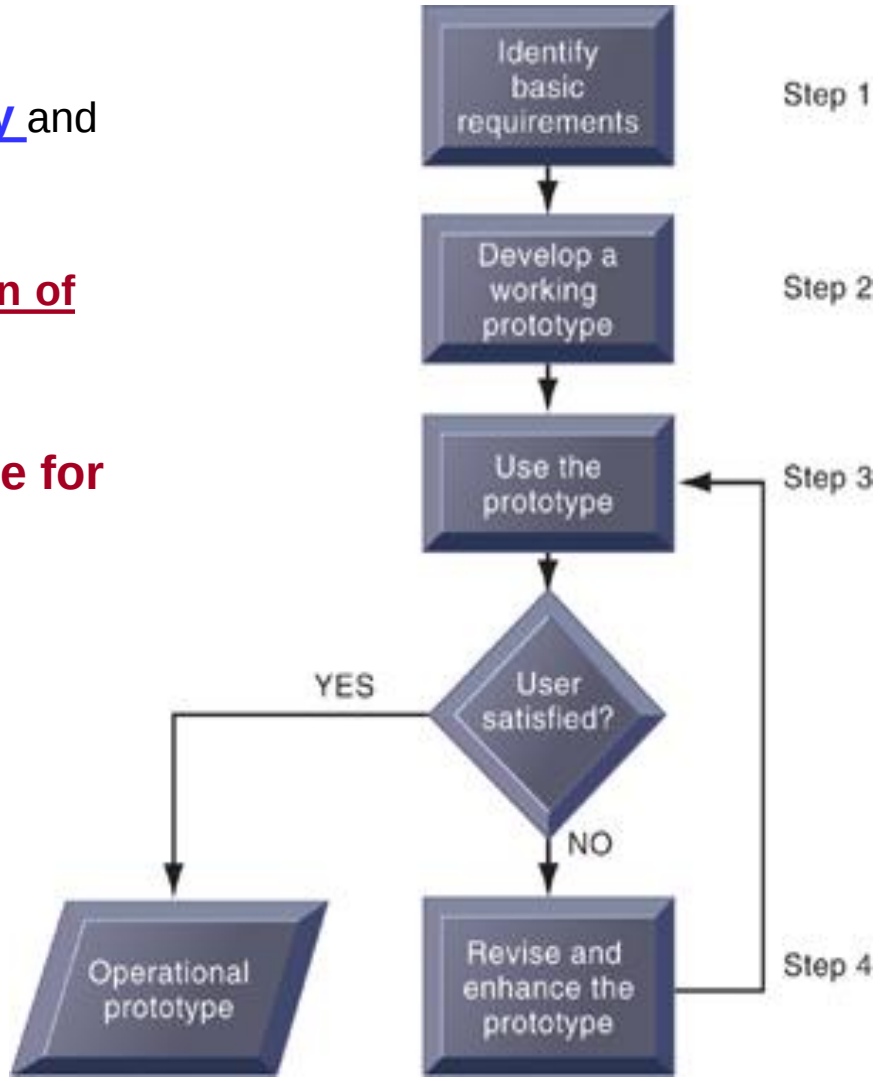
- Must wait a long time before there is “visible” evidence of the new system
- Takes a long time from start to finish

2- Prototyping

- **Building experimental system rapidly** and **inexpensively** for **end users to evaluate**
- **Prototype:** Working but **preliminary version of information system**
- **Approved prototype** serves as **template for final system**

Steps in prototyping

1. Identify user requirements
2. Develop initial prototype
3. Use prototype
4. Revise and enhance prototype



German University in Cairo
الجامعة الألمانية بالقاهرة

2- Prototyping

- **Advantages**
  - Useful if some **uncertainty in requirements** or design solutions
  - Often **used for end-user interface design**
  - More likely to **fulfill end-user requirements**

- **Disadvantages**
  - **May not accommodate large quantities** of **data or large number of users**
  - May **not undergo full testing** or documentation

Prototyping Methodology Assessment

Strengths

- Users get to work with prototype very quickly
- Feedback cycles let users identify changes and refine real requirements

Weaknesses

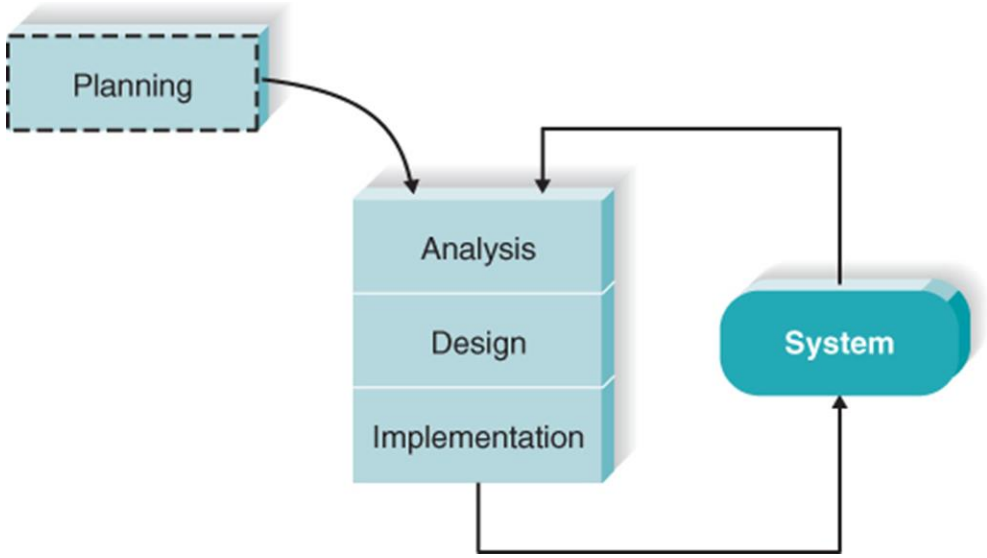
- Superficial analysis may cause problems
- Initial design decisions may be poor
- Overlooked features may be hard to add later

Agile Software Development

Has become popular to describe new approaches that focus on **close collaboration between programming teams** and business experts

Agile Project Management

- Agile means being able to **move quickly and easily**
- Agile today means using a method based on iterative and incremental development, in which requirements and solutions **evolve through collaboration**.



Agile Software Development

http://agilemanifesto.org/

- **Individuals & interactions**
  over processes and tools
- **Working software**
  over comprehensive documentation
- **Customer collaboration**
  over contract negotiation
- **Responding to change**
  over following a plan

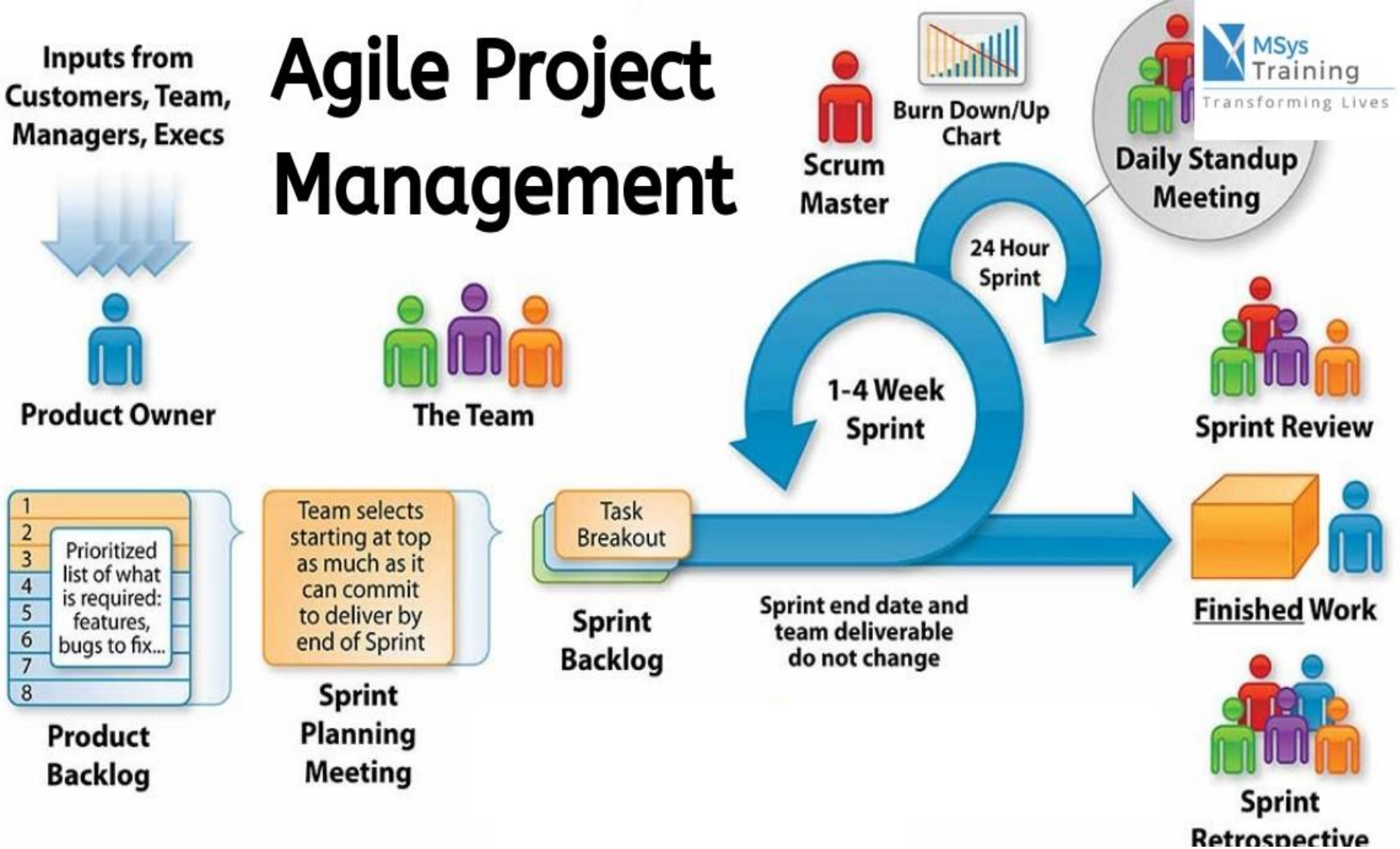
Agile Development Methodology

- Agile SD methods include:
  - Scrum Industry Standard
  - Extreme Programming (XP) Older

Scrum*

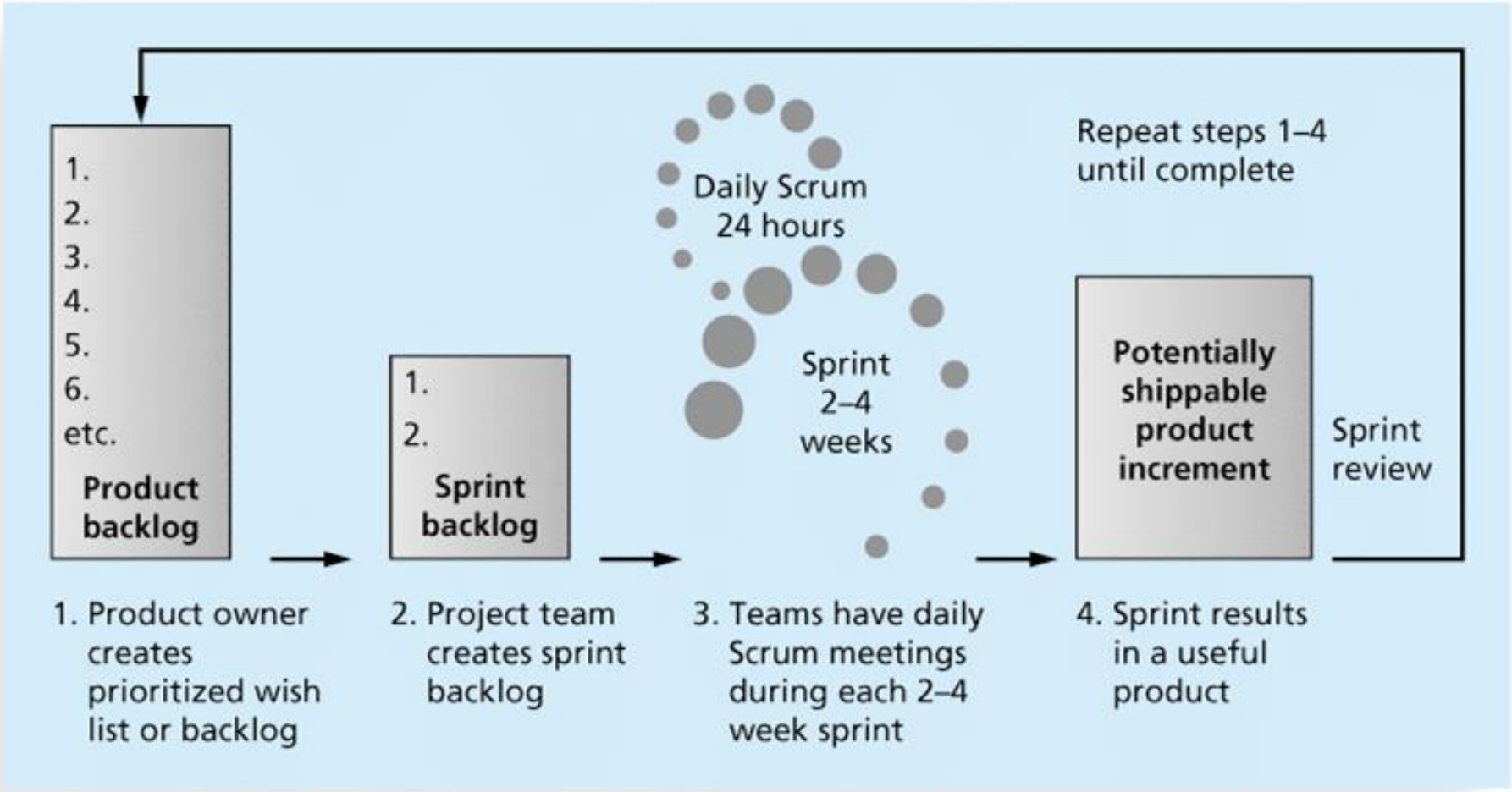
- The leading agile development method for completing projects with a **complex, innovative scope of work** (according to the Scrum Alliance)
- **Scrum** is a framework for project management that emphasizes **teamwork**, accountability and **iterative progress** toward a well-defined goal.
- The framework begins with a **simple** premise: **Start with what can be seen or known**.

* The term was coined in 1986 in a Harvard Business Review study that compared high-performing, cross-functional teams to the scrum formation used by rugby teams.



In **project management**, a '**sprint**' refers to a **set period of time** during which a certain task or activity is completed and then reviewed. **Sprint** is one time-boxed iteration of a continuous development cycle. Within a **Sprint**, planned amount of work has to be completed by the team and made ready for review. ... **Sprint** literal meaning is a short race at full speed. Accordingly, teams usually define a short duration of a **Sprint** up to 2-4 weeks.

Scrum Framework

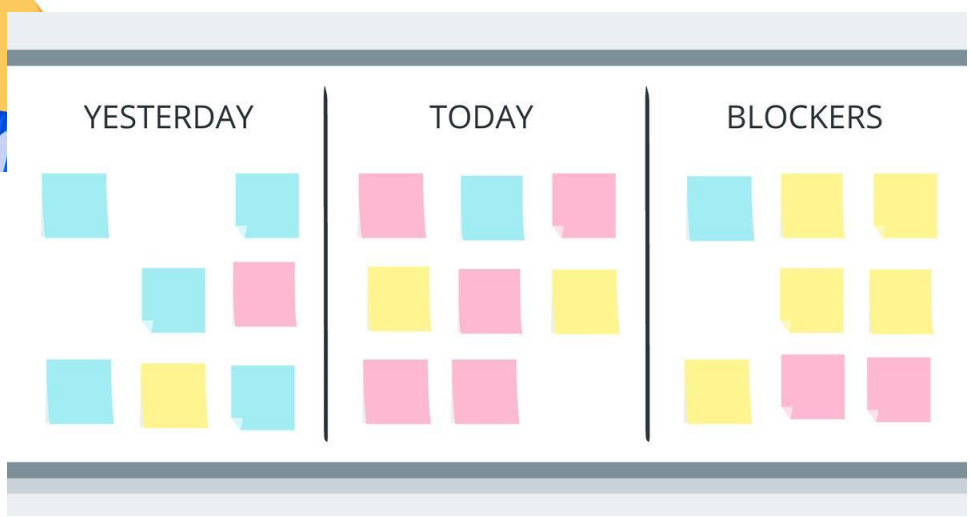


Scrum: main roles

- **Scrum Master:** maintains the processes for a sprint
  - Under supervision of the Project Manager, could be technical lead!
- **Product Owner:** represents the business (and/or stakeholders) Prioritize backlog items
- **Team:** a cross-functional group of people who do the actual analysis, design, implementation, testing, etc.

… and customers, vendors, managers: people for whom the software is being built

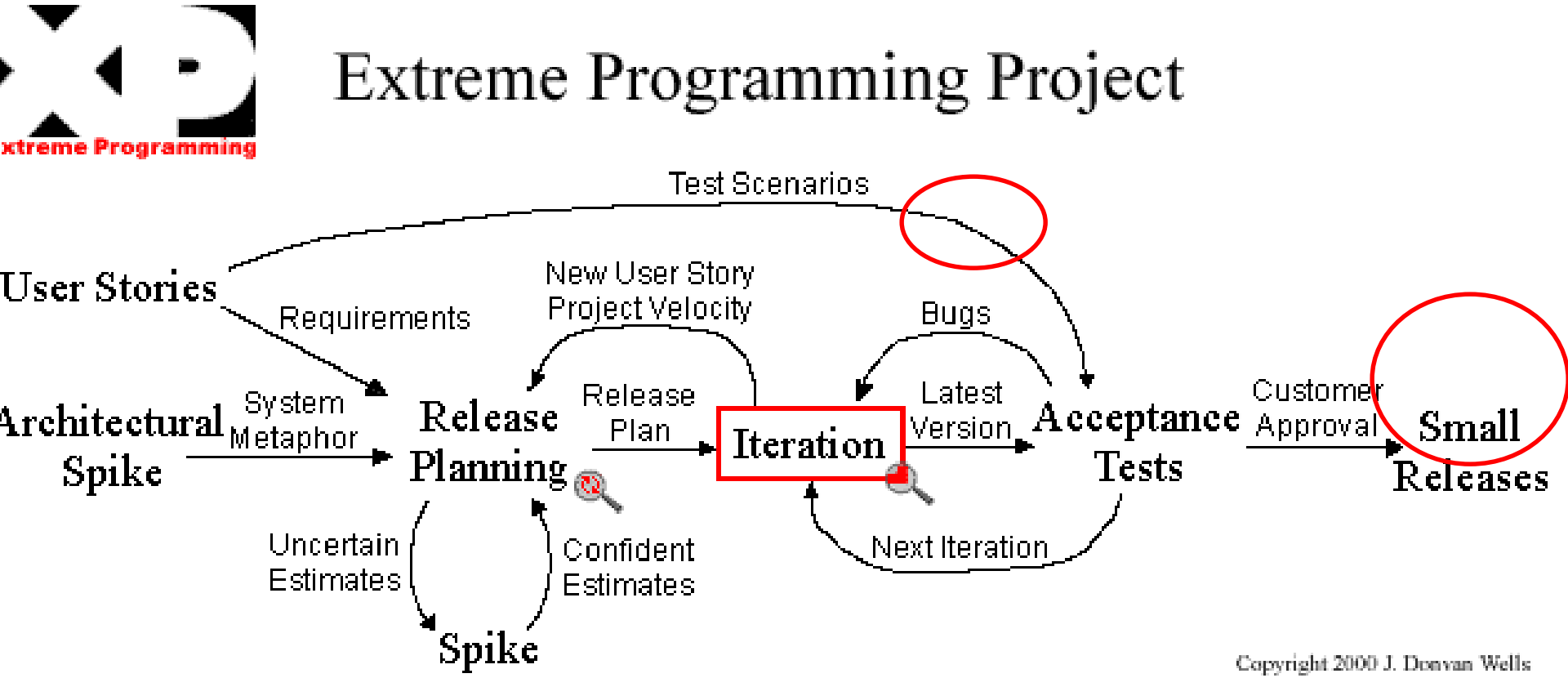
Scrum Standup Meeting



Scrum Standup Meeting

eXtreme Programming

Extreme Programming (XP): XP is an agile software development methodology focused on improving software quality and responsiveness to changing customer requirements. It emphasizes practices such as continuous testing, frequent releases, and close collaboration between developers and customers.



Agile Methodologies Assessment

Strengths

- Fast delivery of results
- Works well in projects with undefined or changing requirements

Weaknesses

- Requires discipline
- Significant user involvement is essential
- Initial high learning curve
- Works best in smaller projects

Agile Versus Waterfall-Based Methodologies

- Agile development approaches have existed for several decades
- Created in part because of dissatisfaction with the sequential, inflexible structure of waterfall-based approaches
- Presently, agile development has made inroads into software development organizations, and studies show an even split between agile and waterfall users

When to use which SW DEV Approach?

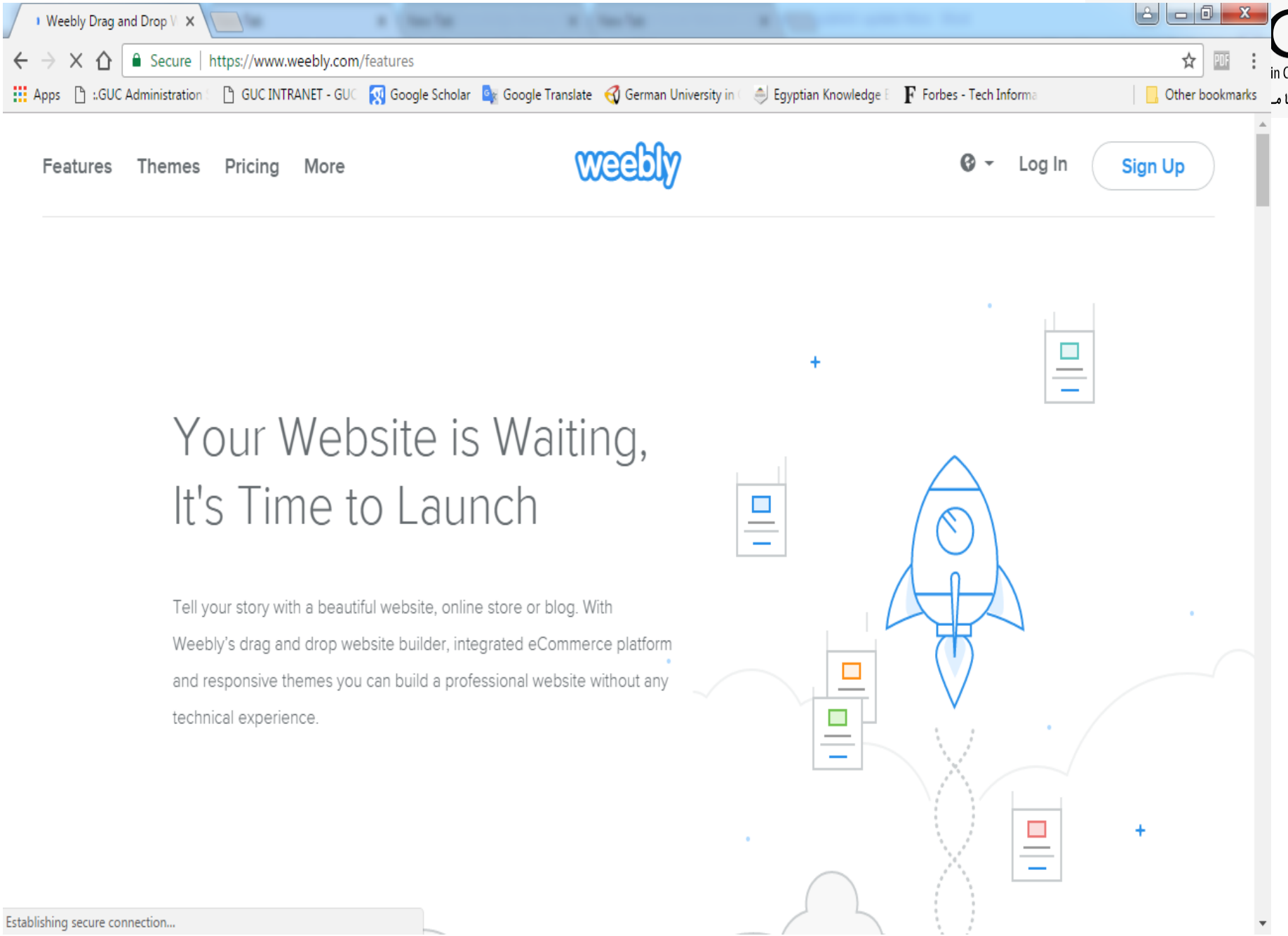
Traditional Waterfall Approach	Modern Agile / Iterative Approach
Repetitive projects with clear well-defined requirements	New projects with risky or unproven requirements
To control scope and limit changes	To have flexible scope
Customer payment based on deliverables	Customer payment based on man-hours or labour
Fixed time projects that cannot exceed a specific duration	Flexible duration projects based on customer needs and requirements

3- End-user development:

- Uses fourth-generation languages to allow **end-users to develop systems with little or no help from technical specialists**
  - **E.g., WYSIWYG (drag-and-drop)**

Categories of fourth generation languages

- **PC software tools** (MS word, MS Access)
- **Query language** (SQL)
- **Report generator** (Crystal Reports)
- **Graphic generator**
- **Application generator** (Web focus)


Weebly Drag and Drop
Secure
https://www.weebly.com/features
Apps
:.GUC Administration
GUC INTRANET - GUC
Google Scholar
Google Translate
German University in
Egyptian Knowledge
Forbes - Tech Informa
Other bookmarks
Features
Themes
Pricing
More
weebly
Log In
Sign Up
Your Website is Waiting,
It's Time to Launch
Tell your story with a beautiful website, online store or blog. With Weebly's drag and drop website builder, integrated eCommerce platform and responsive themes you can build a professional website without any technical experience.
Establishing secure connection...

German University in Cairo
الجامعة الألمانية بالقاهرة

3- End-user development:

Advantages:

- **Less procedural** than **conventional programming** languages
- More **rapid completion of projects**, high-level of **user satisfaction**

Disadvantages:

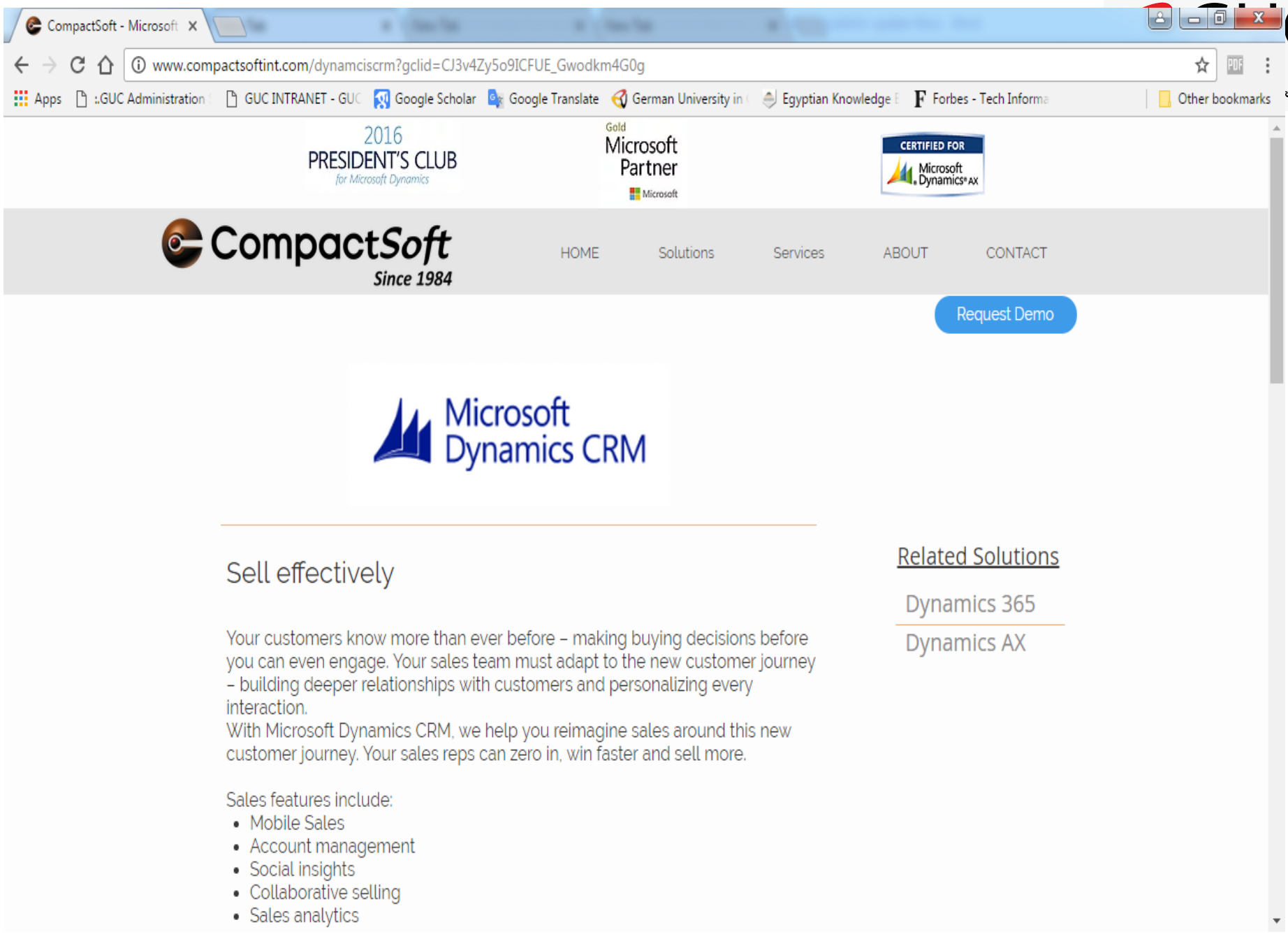
- Not **designed for processing-intensive applications**, **inadequate control**, **testing**, **documentation**, or **adherence to standards**

4- Alternative Systems Building Approaches - a) Purchase Software Package (& possibly customize it)

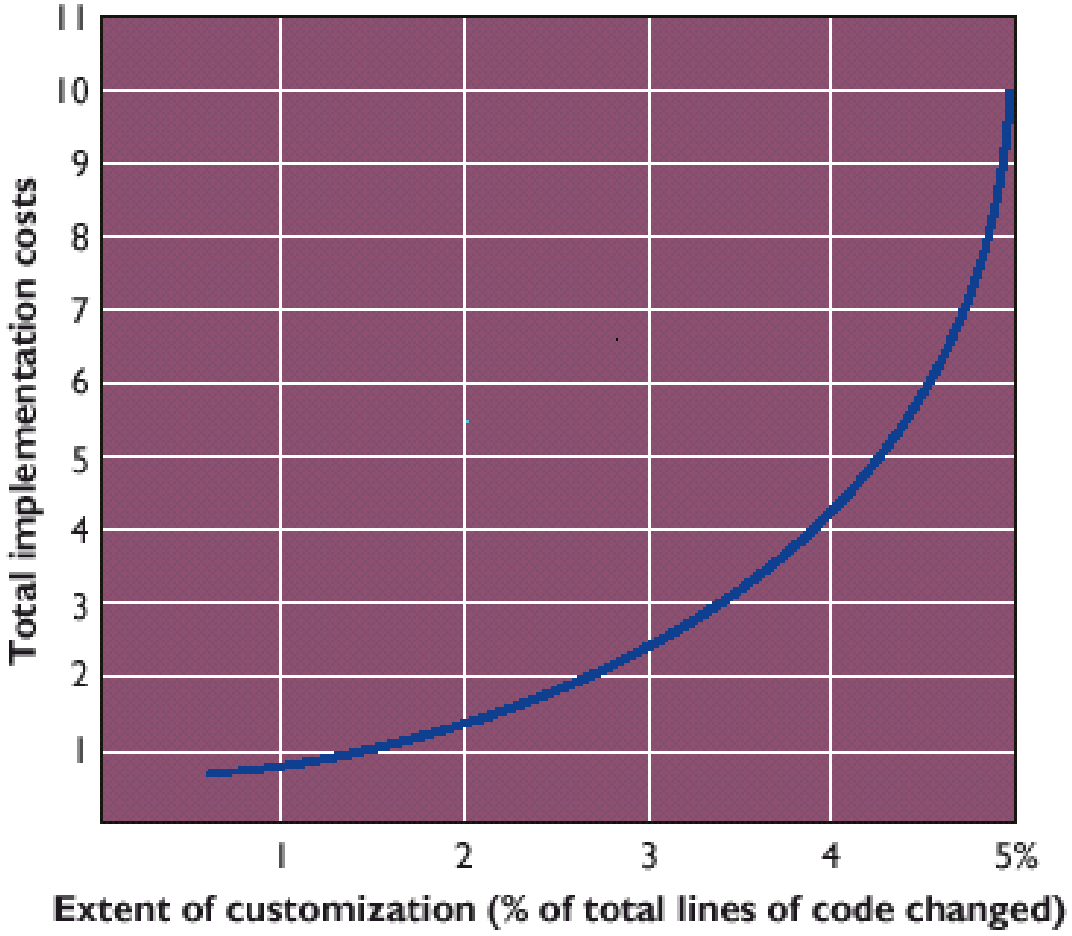
- **Application software**
  - **Set of prewritten, precoded application software programs commercially available for sale or lease**
  - Many **applications are common** to all business organizations
    - **Examples**: **payroll**, **general ledger**, **inventory control, (ERP)**

Advantages

- Save **time** and **money**
- Many packages **offer customization features**:
  - Allow **software package to be modified to meet unique requirements** without destroying integrity of package software


CompactSoft - Microsoft
www.compactsoftint.com/dynamciscrm?gclid=CJ3v4Zy5o9ICFUE_Gwodkm4G0g
Apps
:.GUC Administration
GUC INTRANET - GUC
Google Scholar
Google Translate
German University in
Egyptian Knowledge
Forbes - Tech Informa
Other bookmarks
2016
PRESIDENT'S CLUB
for Microsoft Dynamics
Gold
Microsoft
Partner
Microsoft
CERTIFIED FOR
Microsoft Dynamics AX
CompactSoft
Since 1984
HOME
Solutions
Services
ABOUT
CONTACT
Request Demo
Microsoft
Dynamics CRM
Sell effectively
Your customers know more than ever before – making buying decisions before you can even engage. Your sales team must adapt to the new customer journey – building deeper relationships with customers and personalizing every interaction.
With Microsoft Dynamics CRM, we help you reimagine sales around this new customer journey. Your sales reps can zero in, win faster and sell more.
Sales features include:
• Mobile Sales
• Account management
• Social insights
• Collaborative selling
• Sales analytics
Related Solutions
Dynamics 365
Dynamics AX

The Effects of Customizing a Software Package on Total Implementation Costs



4- Alternative Systems Building Approaches
b) Custom/in-house Development

Advantages

- Get exactly what we want
- New system built consistently with existing technology and standards
- Build and retain technical skills and functional knowledge in-house
- Allows team flexibility and creativity
- Unique solutions created for strategic advantage

Disadvantages

- Requires significant time and effort
- May add to existing backlogs
- May require skills we do not have
- Often costs more
- Often takes more calendar time
- Risk of project failure

4-Alternative Systems Building Approaches - c) Outsourcing

Several types of outsourcing

- **Application service providers (ASPs)**
  - **Subscribing companies use software and computer hardware provided by ASP** as **technical platform for systems. (SaaS).**
  - **E.g. CRM online**

- **Domestic or foreign external (offshore) vendors**
  - **Hired** to design, create software **(the company would operate the system on its own computers)**

Alternative Systems-Building Approaches

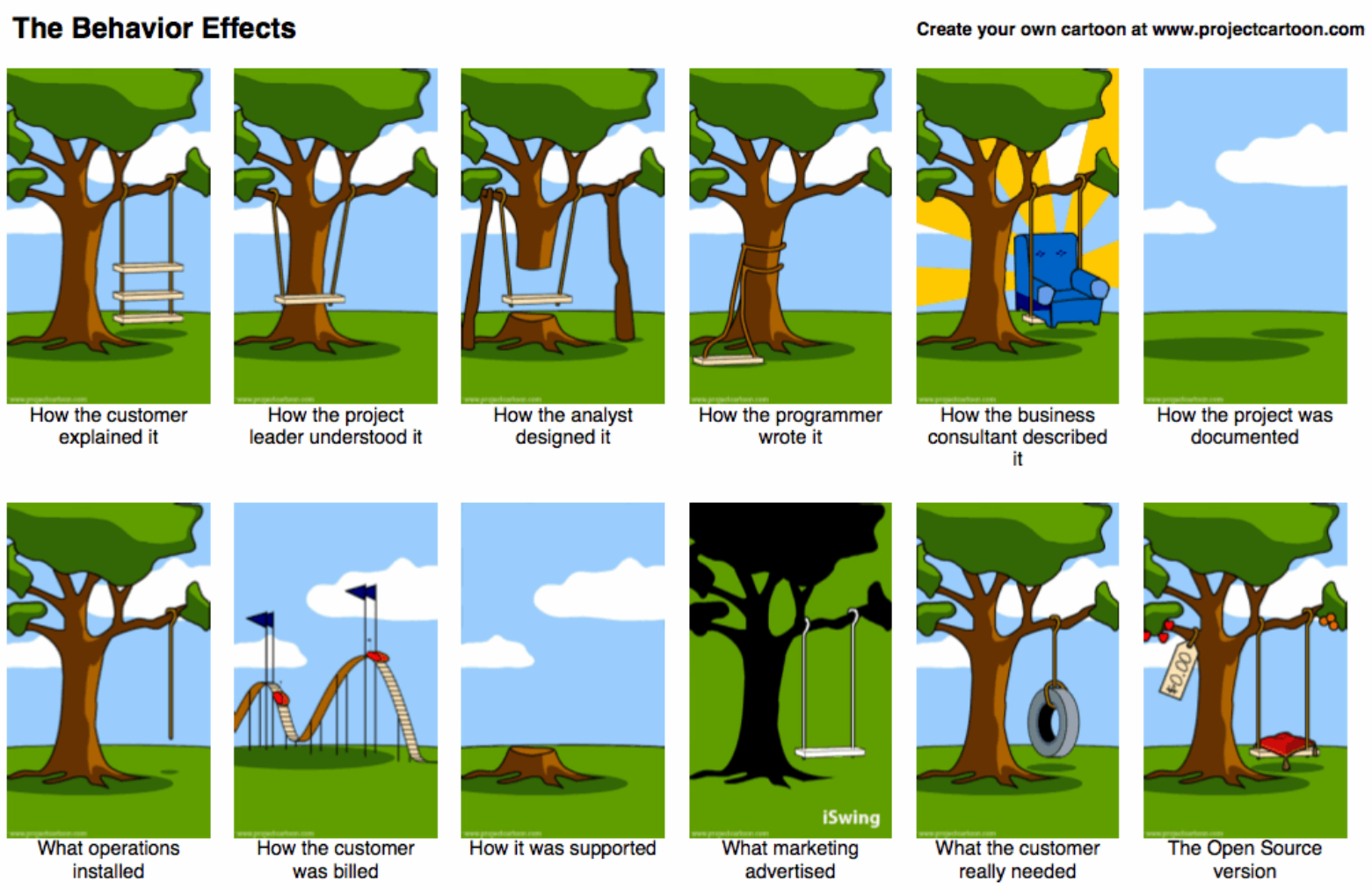
Advantages

- Allows organization **flexibility in IT needs**
- Allows vendors:
  - **Economies** of scale
  - Enhance **core competencies**

Disadvantages

- **Hidden costs**, loss of control
- For example, the case of **offshore outsourcing**

The Dilemma


The Behavior Effects
Create your own cartoon at www.projectcartoon.com
How the customer explained it
How the project leader understood it
How the analyst designed it
How the programmer wrote it
How the business consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
iSwing
What marketing advertised
What the customer really needed
The Open Source version

Outline

1. Systems Development Project
2. Traditional Systems Development Approaches
3. Alternative Systems Development Approaches
4. **Conclusion**

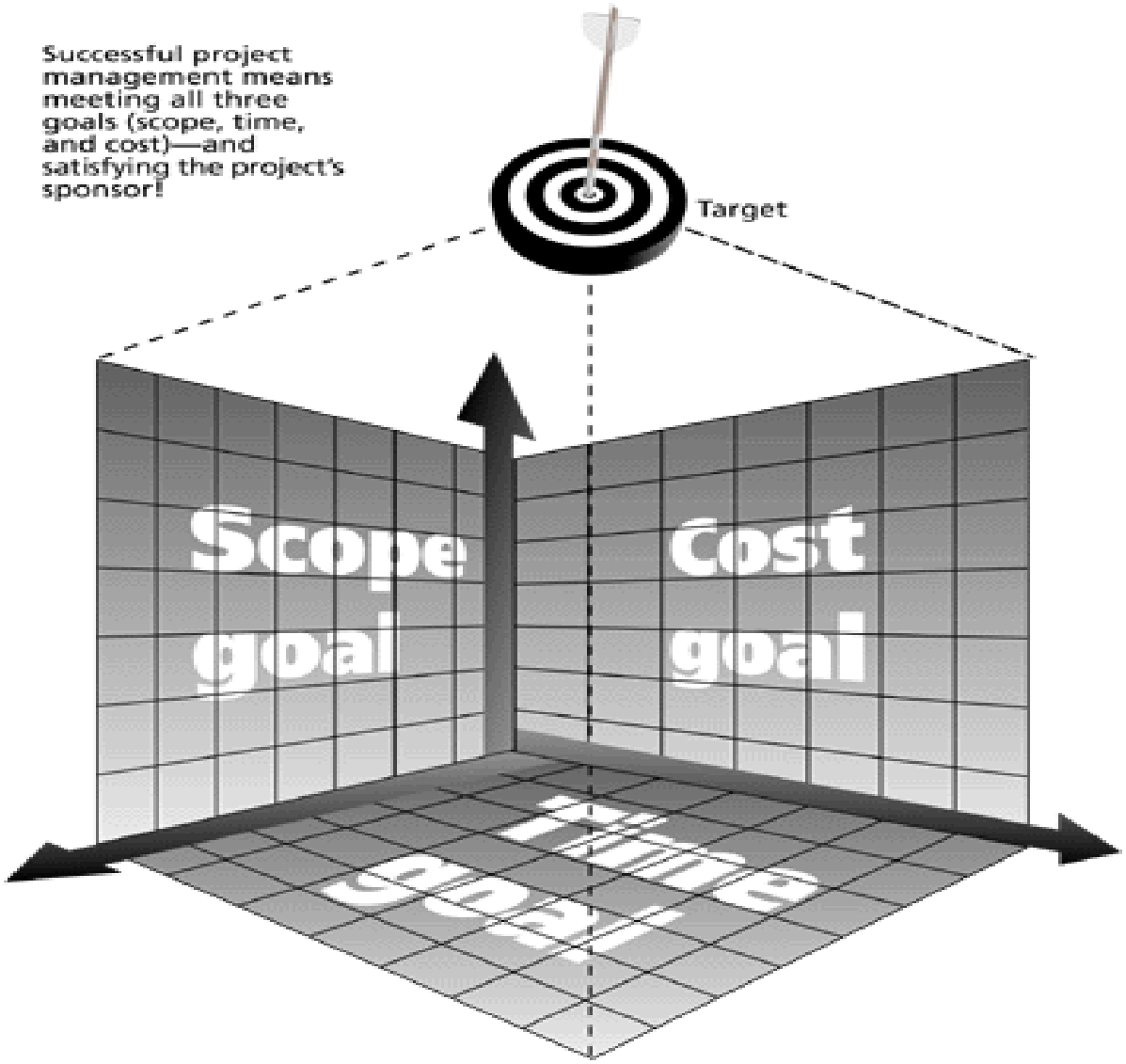
The Triple Constraint

Every project is constrained in different ways by its

1. **Scope** goals:
   What is the project trying to accomplish?
2. **Time** goals:
   How long should it take to complete?
3. **Cost** goals:
   What should it cost?

It is the **project manager's duty** to balance these three often competing goals

Triple Constraint of Project Management



Which SW Dev Approach for this problem …?

1. We need to implement computer networks for a client. The hardware and software is the same and we implemented in 100 projects before.

1. We need to implement a new e-Business website from scratch to sell our products online. This is the first time we implement such technology and projects, we need to make it in steps (module-by-module).

1. We need to automate the production line with sensors to prevent errors and take corrective actions automatically. We are unsure of the exact sensors and software we will need or what could go wrong?!

1. We want to implement a new point-of-sale system at a supermarket for each of the following:
   a) The most flexible terms of contract where we can change the terms easily
   b) The cheapest and quickest approach for me as a client without being charged surprising fees (want to control costs)
   c) A clear and fixed contract from the beginning that doesn't change from any party
   d) I want to implement this based on a cost-plus contract where I pay for the materials and wages for whatever is consumed in the project and not on a whole bulk agreed term.

THANK YOU FOR YOUR ATTENTION

NEXT WEEK: Project Integration and procurement management

NEXT TUTORIAL: IT Project Success / Failure Factors Case Study

+ Microsoft Project Introduction (Lab)