ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN
Vietnam - Korea University of Information and Communication Technology

Software Project Management

Methodologies

Lecture Objectives

- Plan-Driven versus Agile Methodologies
- Plan-Driven Methodologies (Examples)
  - Waterfall
  - Spiral
- Agile Methodologies (Examples)
  - XP
  - Scrum
- Selecting a Methodology for your Project
- Collecting and Analyzing Data for Different Methodologies
- Plan-Driven and Agile Misconceptions

Plan Driven vs Agile Methodologies

- Plan-Driven methodologies assert the need for strong process discipline and rigorous practices.
- Quality products come from quality processes.
- Assumes requirements are known at start.
- “Here is the schedule & budget given these requirements.”
- Change managed formally.
- Rework is bad.

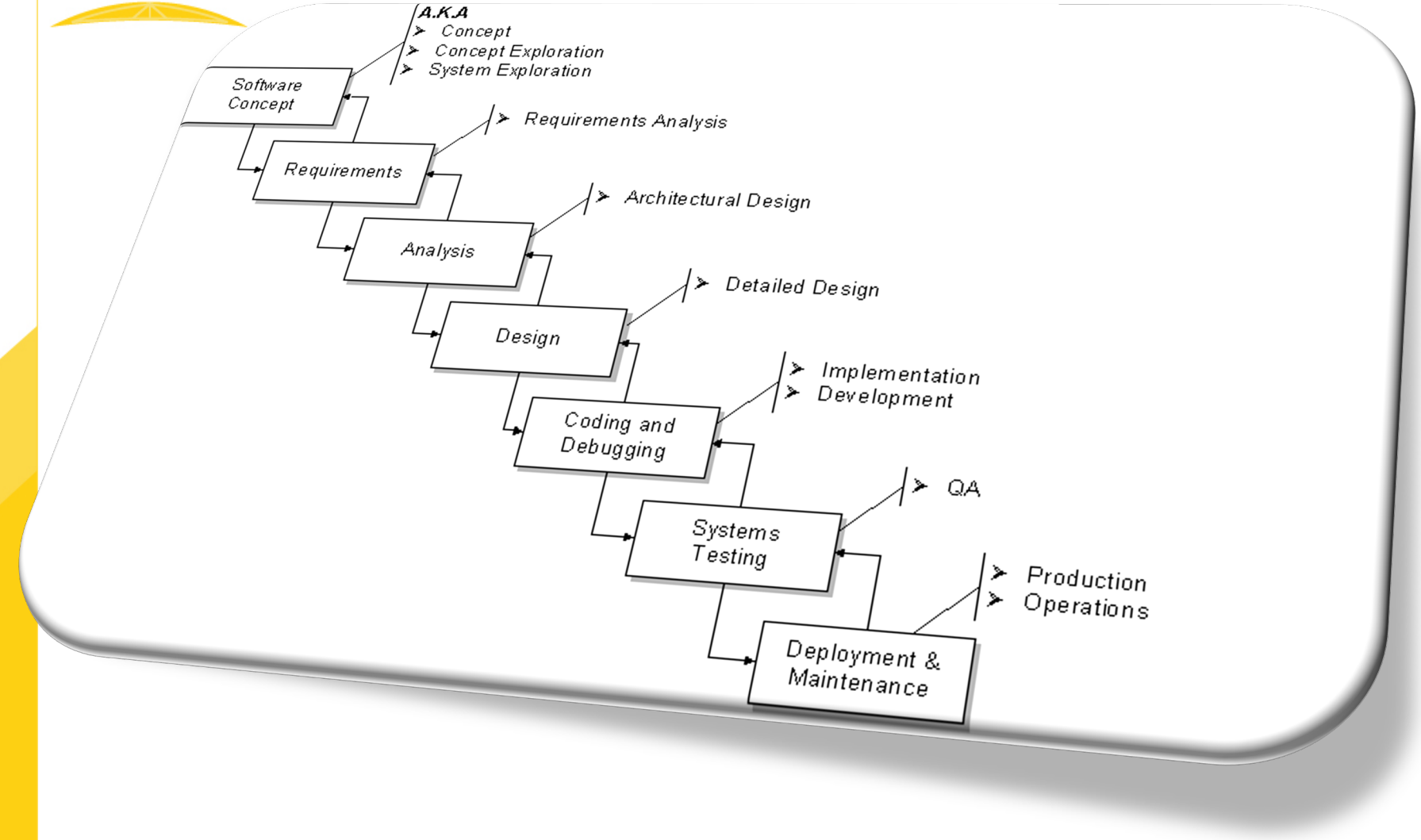
Plan Driven vs Agile Methodologies

- Agile methodologies use lighter, more adaptive paradigms.
  - Product built so you always have a working system and can “release” at any time.
  - Refactoring is good.
- The Agile manifesto says that we have come to value:
  - Individuals & interaction over process & tools.
  - Working software over comprehensive docs.
  - Customer collaboration over contract negotiation.
  - Responding to change over following a plan.
  - That is, while there is value in the items on the right, we value the items on the left more.

Plan Driven vs Agile Methodologies

- Which one is best? Depends on a project's attributes and environment.
  - Requirements emerging or known?
  - Project and organization's culture.
  - Team's expertise.
  - Customer's availability.
  - Nature of relationship with customer.
  - Sophistication of customer.

Waterfall


A.K.A
Concept
Concept Exploration
System Exploration
Software Concept
Requirements Analysis
Requirements
Architectural Design
Analysis
Detailed Design
Design
Implementation
Development
Coding and Debugging
QA
Systems Testing
Production
Operations
Deployment & Maintenance

Waterfall

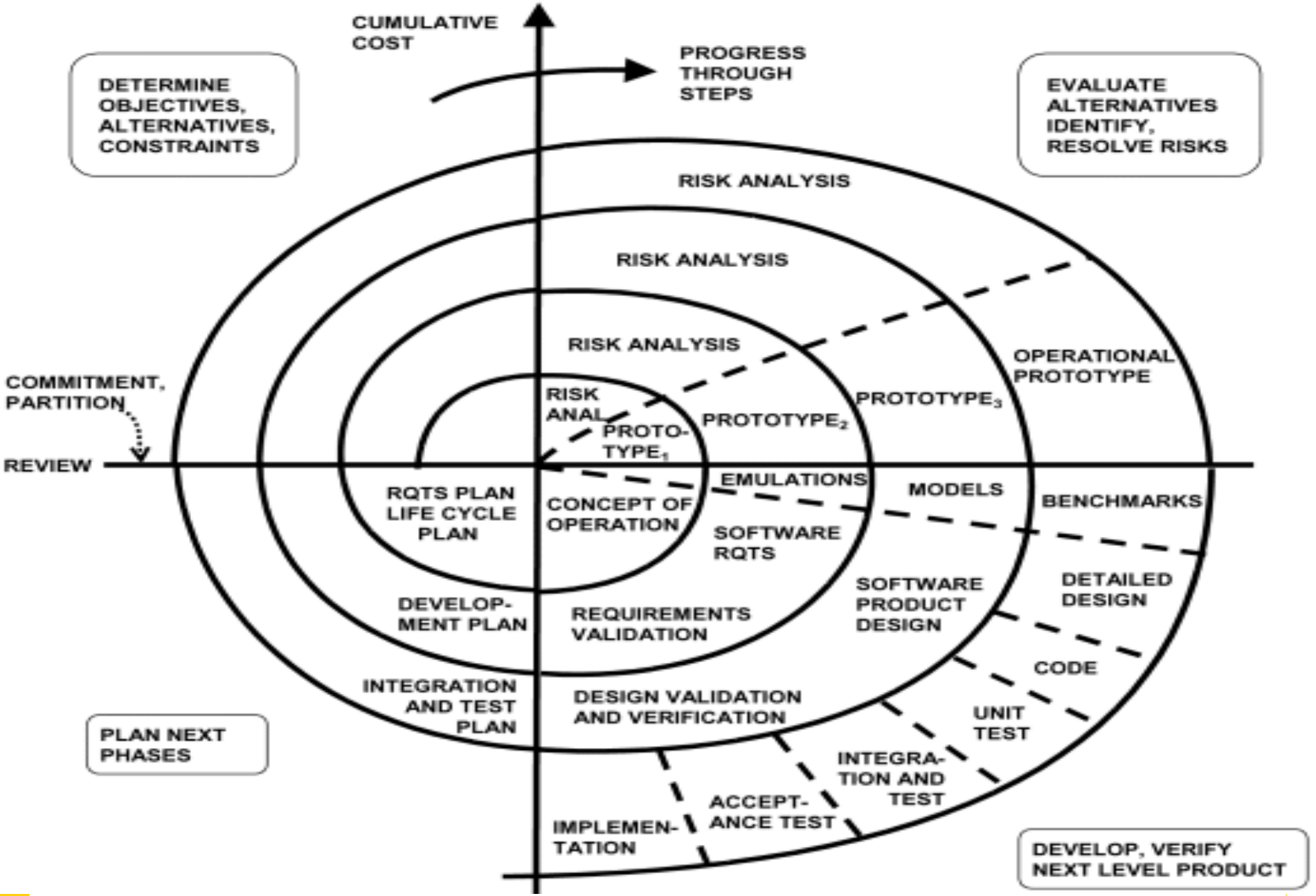
- The “granddaddy” of models
- Linear sequence of phases
  - “Pure” model: no phases overlap
- Document driven
- All planning done up-front

Waterfall Strengths

- Works well for projects with:
  - Stable product definition
  - Well-understood technologies
  - Quality constraints stronger than cost & schedule
  - Supports technically weak staff
    - Provides structure
    - Good for overseas projects

Waterfall Disadvantages

- Not flexible:
  - Rigid march from start->finish
  - Teams can end up following obsolete plans
- Difficult to fully define requirements up front.
- Can be an over reliance on process.
- Can have processes that aren't actually followed.
- Can waste a lot of time producing excessive documentation.
- Few visible signs of progress until the end.
- Customer/User doesn't see product until the end.


CUMULATIVE COST
PROGRESS THROUGH STEPS
DETERMINE OBJECTIVES, ALTERNATIVES, CONSTRAINTS
EVALUATE ALTERNATIVES IDENTIFY, RESOLVE RISKS
RISK ANALYSIS
RISK ANALYSIS
RISK ANALYSIS
COMMITMENT, PARTITION
REVIEW
RISK ANAL
PROTO-TYPE1
PROTOTYPE2
PROTOTYPE3
OPERATIONAL PROTOTYPE
RQTS PLAN LIFE CYCLE PLAN
CONCEPT OF OPERATION
EMULATIONS
SOFTWARE RQTS
MODELS
BENCHMARKS
DEVELOP-MENT PLAN
REQUIREMENTS VALIDATION
SOFTWARE PRODUCT DESIGN
DETAILED DESIGN
CODE
INTEGRATION AND TEST PLAN
DESIGN VALIDATION AND VERIFICATION
UNIT TEST
INTEGRA-TION AND TEST
ACCEPT-ANCE TEST
IMPLEMEN-TATION
PLAN NEXT PHASES
DEVELOP, VERIFY NEXT LEVEL PRODUCT

Spiral

- A Series of Mini-projects, where each addresses a set of “risks”.
  - Start small, explore risks, prototype, plan, repeat
- Emphasizes risk analysis & management in each phase.
- Early iterations are “cheapest”.
- Number of spirals (iterations) is variable.

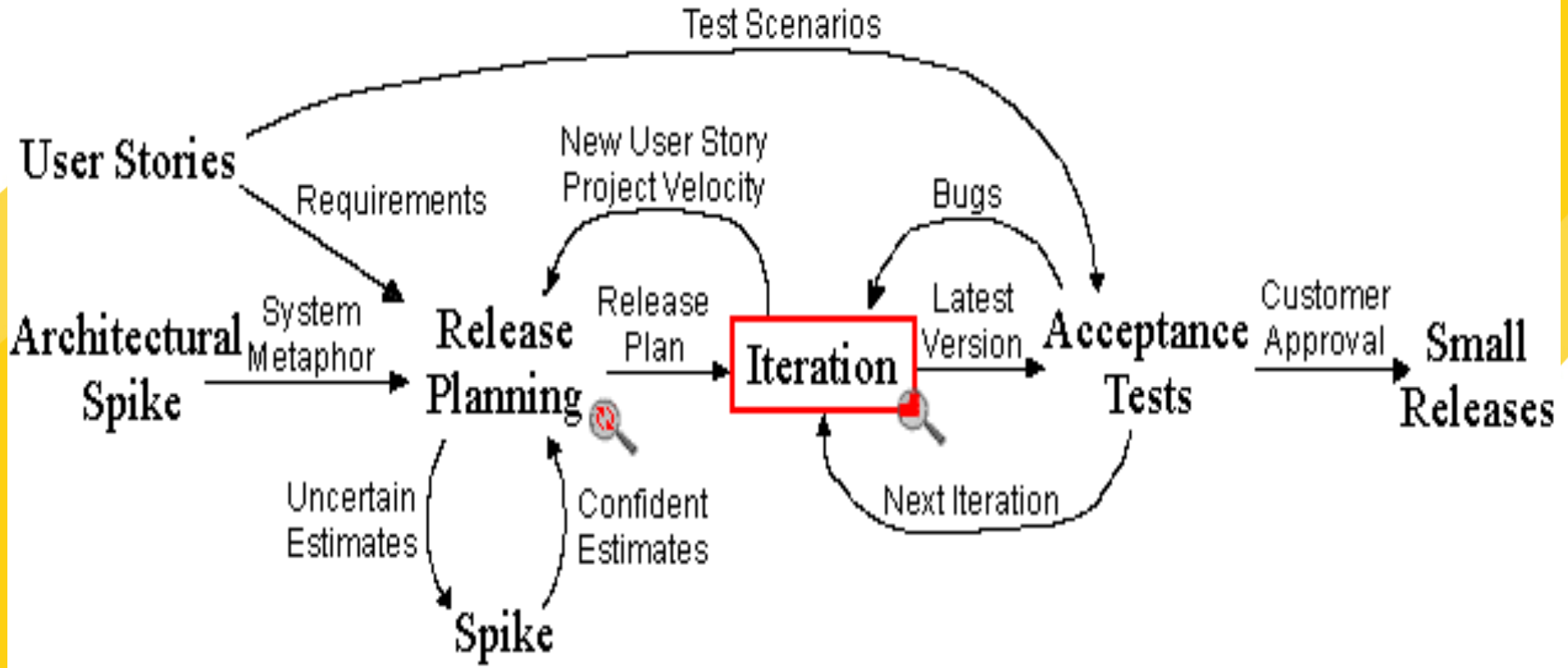
Spiral Strengths

- Advantages
  - Can be combined with other models
  - As costs increase, risks decrease
  - Risk orientation provides early warning

Spiral Disadvantages

- Disadvantages
  - Complicated
  - Requires conscientious, knowledgeable management

eXtreme Programming (XP)



eXtreme Programming (XP)

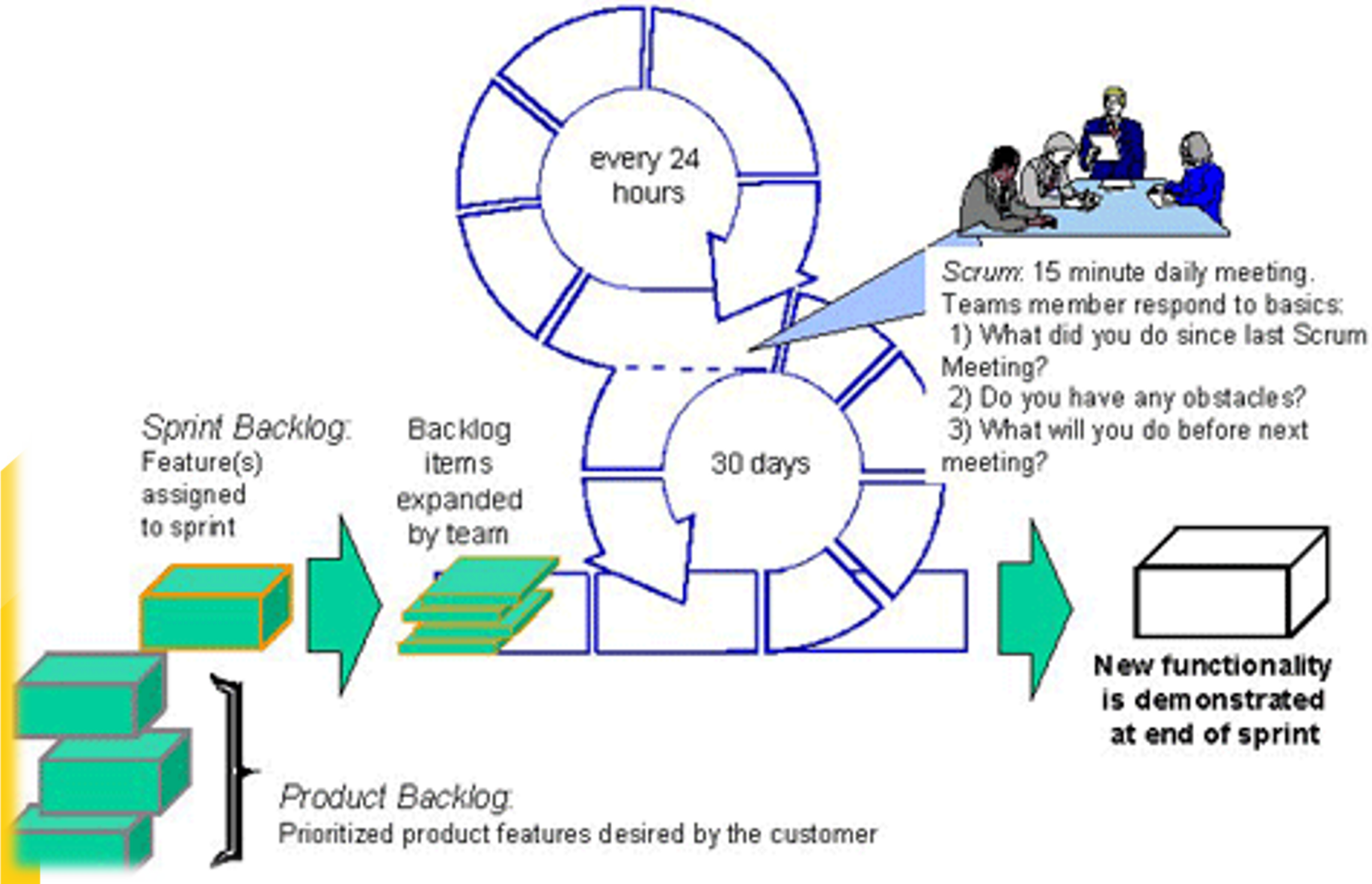
- Based on four values:
  - Communication: Implement practices that force communication in a positive fashion.
  - Simplicity: Build the simplest product that meets the customer's needs.
  - Feedback: Team must obtain and value feedback from customer, system and each other.
  - Courage: Be prepared to make hard decisions.
- Small teams.
- Customer is continuously present on-site.
- Short iterations (no more than 3 weeks).
- Test first approach.
- Continuous build and integration.

XP Strengths

- Most widely recognized Agile method.
- Strong track record of success with small applications.
- Very strong communications and involvement with customer mitigates risk of emerging requirements.

XP Disadvantages

- Scaling is an issue
  - Little experience with larger teams
  - Tends to need more disciplined management methods for larger teams to work
- Some barriers to use:
  - Teams that are not collocated
  - Long feedback cycles (customer not available in a timely manner)


every 24 hours
30 days
Scrum: 15 minute daily meeting.
Teams member respond to basics:
1) What did you do since last Scrum Meeting?
2) Do you have any obstacles?
3) What will you do before next meeting?
Sprint Backlog:
Feature(s) assigned to sprint
Backlog items expanded by team
New functionality is demonstrated at end of sprint
Product Backlog:
Prioritized product features desired by the customer

Scrum

- Based on the concept that software development is not a defined process, but rather an empirical process that may or may not be repeated based on circumstances.
- Self organizing teams.
- Has a definite project management emphasis.
  - Managed by Scrum Master
  - Based on a backlog on product enhancements
  - 30-day ‘sprints’
  - Pre- and post-sprint activities
  - Short (<30 minutes) daily Scrum meetings to monitor status & communicate problems

Scrum Strengths

- One of the few agile methods that has attempted to scale for larger projects by having scrums of scrum masters.
  - No specific engineering practices prescribed, but many Scrum teams are adopting XP.
- Scrum offers a controlled means for introducing Agile methods into a traditionally plan-driven environment.
- No changes allowed during a sprint!
- Each sprint produces a potentially shippable product increment.

Scrum Disadvantages

- Requires hand-on management, but not micromanagement.
  - Management must be willing to make changes to help Scrum teams succeed.
- Scrum requires constant monitoring both quantitatively and qualitatively.
- Requires management to delegate decision-making authority to the Scrum team.
- Managers must let Scrum teams make their own decisions, even allowing them to fail if necessary.
- Some workers are not comfortable with the
- responsibility Scrum enables.

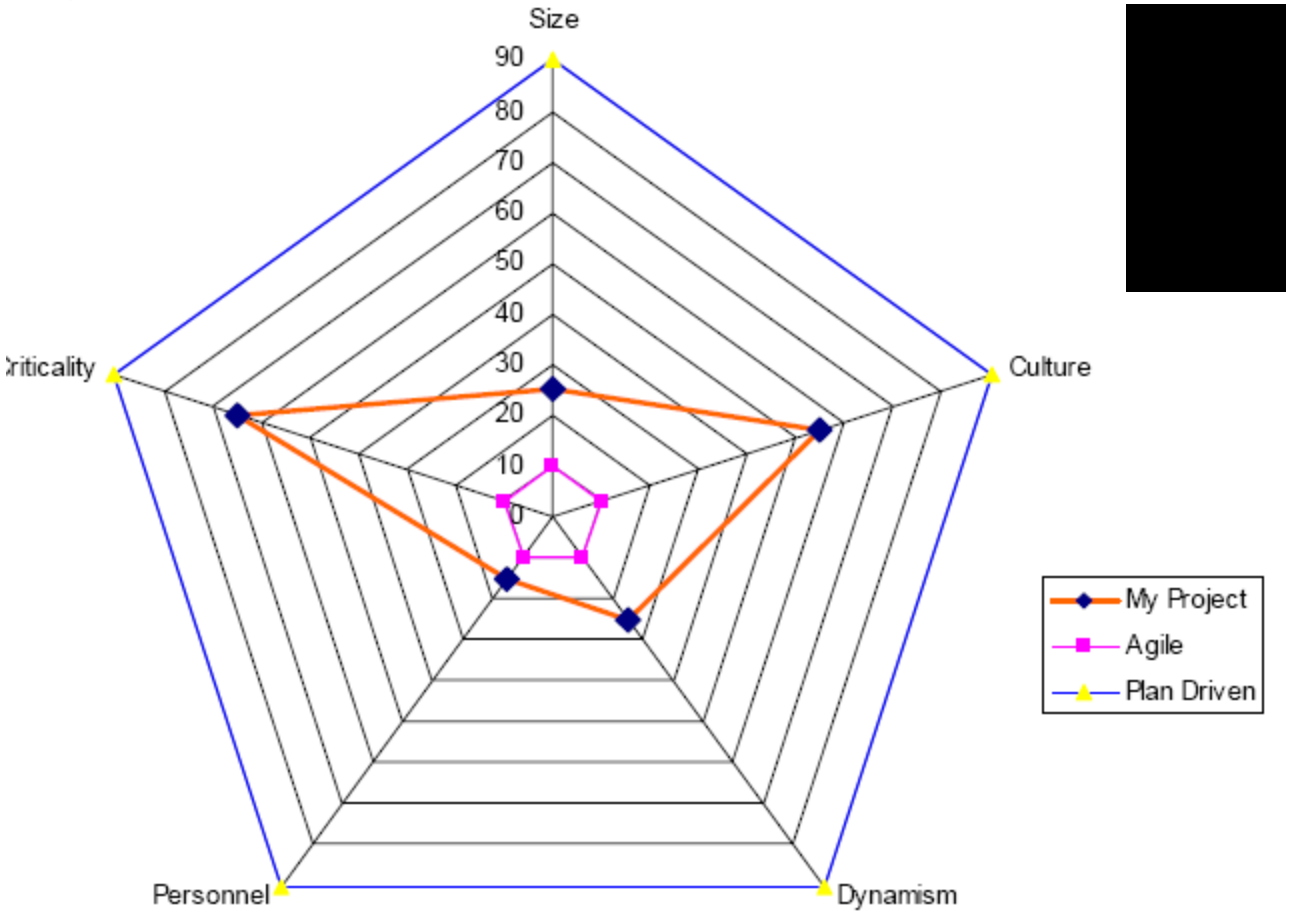
Choosing Your Lifecycle

- Greatly influences your chance of success.
- Not choosing a lifecycle is a bad option.
- Varies by project.
  - How well are requirements understood?
  - What are the risks?
  - Is there a fixed deadline?
  - How experienced is the team or customer?
- See table 7.1 in Rapid Development.

Methodology Home Grounds

Agile & Plan-Driven Home Grounds

(Source: Boehm, Barry W. and Turner, Richard. *Balancing Agility and Discipline.* Boston, MA: Addison-Wesley, 2004)



Methodology Home Ground

- Criticality (of product assurance and safety)
- Agile projects build software quickly and find out from feedback what features or activities will add the most value next.
- Primary goals of plan-driven methods are predictability, stability and high assurance.
- Personnel (developers)
- Current thinking is that Agile methods work best with higher-skilled people.
- Plan-driven methods can incorporate less-capable developers with lower risk.

Methodology Home Ground

- Size
  - Current thinking is that Agile projects work best with small to medium teams working on small applications.
  - Plan-driven methods scale better to large projects.
- Culture
  - Agile is more craft oriented, where the developer is trusted to do whatever is necessary.
  - Plan-driven is based on clear policies and procedures.

Methodology Home Ground

- Dynamism
  - Agile methods are at home with both high and low rates of change.
  - Plan-driven methods prefer low rates of change.

Methodology Data Differences

- Project Tracking
  - Plan Driven: Earned Value.
  - Agile: Burn Down Charts.
- Product Change
  - Plan Driven: Needs to be managed.
  - Agile: Natural part of meeting customer needs.
- Rework and Refactoring
  - Plan Driven: Rework is bad and should be eliminated.
  - Agile: Refactoring is a natural part of meeting customer needs.

Plan Driven Misconceptions

- Plan-driven methods are uniformly bureaucratic.
  - Overly bureaucratic cultures and methods can stultify any software development.
- Having documented plans guarantees compliance with plans.
  - Not necessarily.
- Plan-driven methods can succeed with a lack of talented people.
  - Plan-driven methods can succeed with a smaller % of talented people.
- High maturity guarantees success.
  - Explicit, documented plans provide more of a safety net than tacit plans.
- There are no penalties in applying plan-driven methods when change is unforeseeable.
  - Plan-driven methods work best when accommodating foreseeable change.

Agile Misconceptions

- Agile methods don’t plan.
  - Agile methods do plan.
  - They get much of their speed and agility through creating and exploiting tacit knowledge
- Agile methods require uniformly talented people.
  - Agile methods work best when there is a critical mass of highly talented people involved.
- Agile methods can make the slope of the cost to change vs. time curve uniformly flat.
  - Agile methods can reduce the slope of the cost to change vs. time curve.
- YAGNI is a universally safe assumption, and won’t alienate customers.
  - YAGNI helps handle unforeseeable change, but is risky when change is foreseeable.

Summary

- We've described, in detail, the key differences between projects and software projects.
- Software projects still being poorly led and managed.
- Definition of insanity: "Changing nothing, but expecting different results."
- Possibly your situation is:
  - You're not responsible
  - You're not in charge
  - You don't feel confident …
  - Pick something that you think will make a difference that is within your domain of control, and implement it, or work with your teammates…
  - In a year, what will be better for your projects …

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN
Vietnam - Korea University of Information and Communication Technology

Questions & Answers

Methodologies