Software Process and Project Management

Assoc. Prof. Waraporn Jirapanthong, Ph.D.

Topics

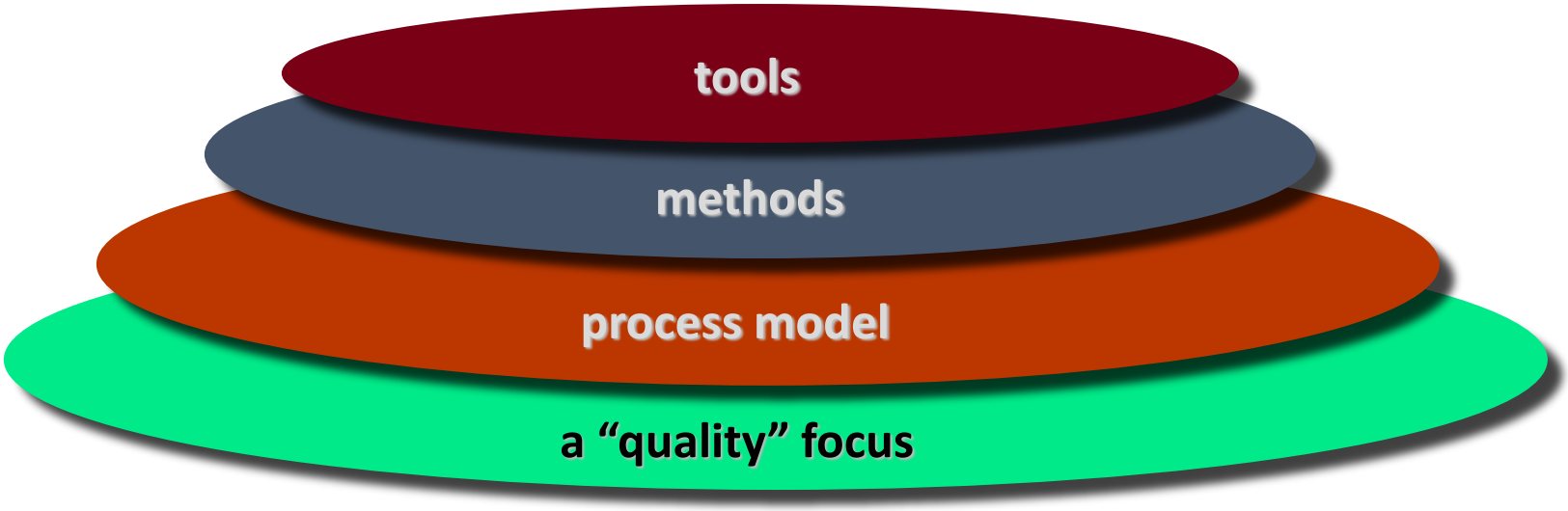
Software Process Model

Software Process

Software Project Management

Software Process Model

A Layered Technology



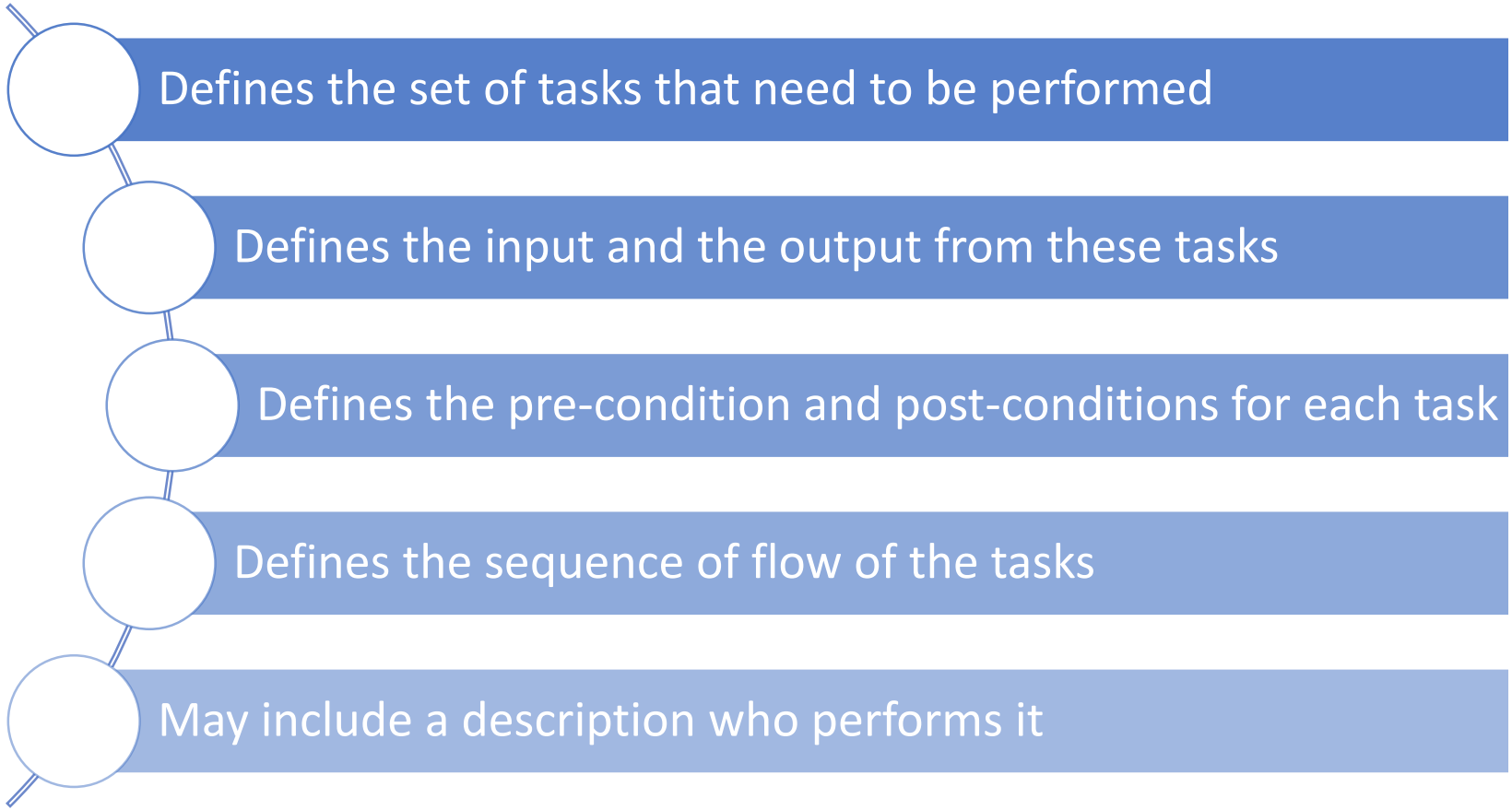
Software Engineering

Why Process Models?

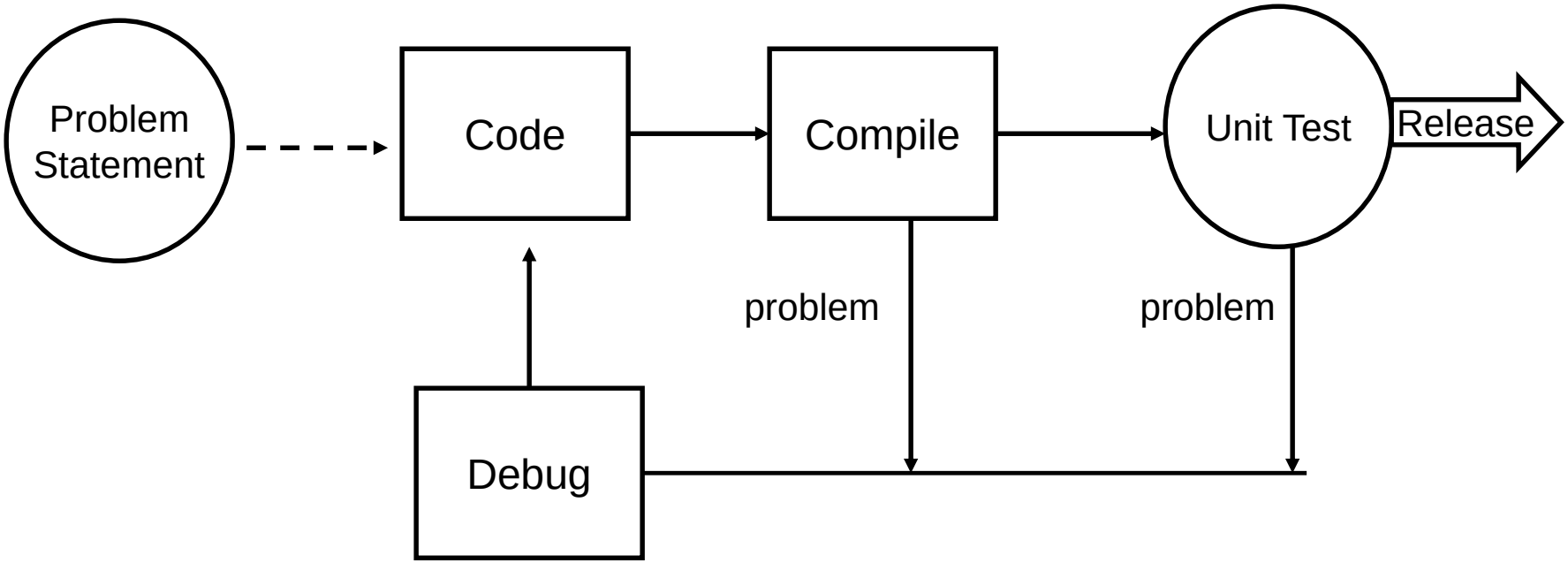
Provide guidance for a systematic coordination and controlling of the tasks and of the personnel who performs the tasks

Note the key words: coordination, tasks, people

Process Model



A Simple and Familiar Process

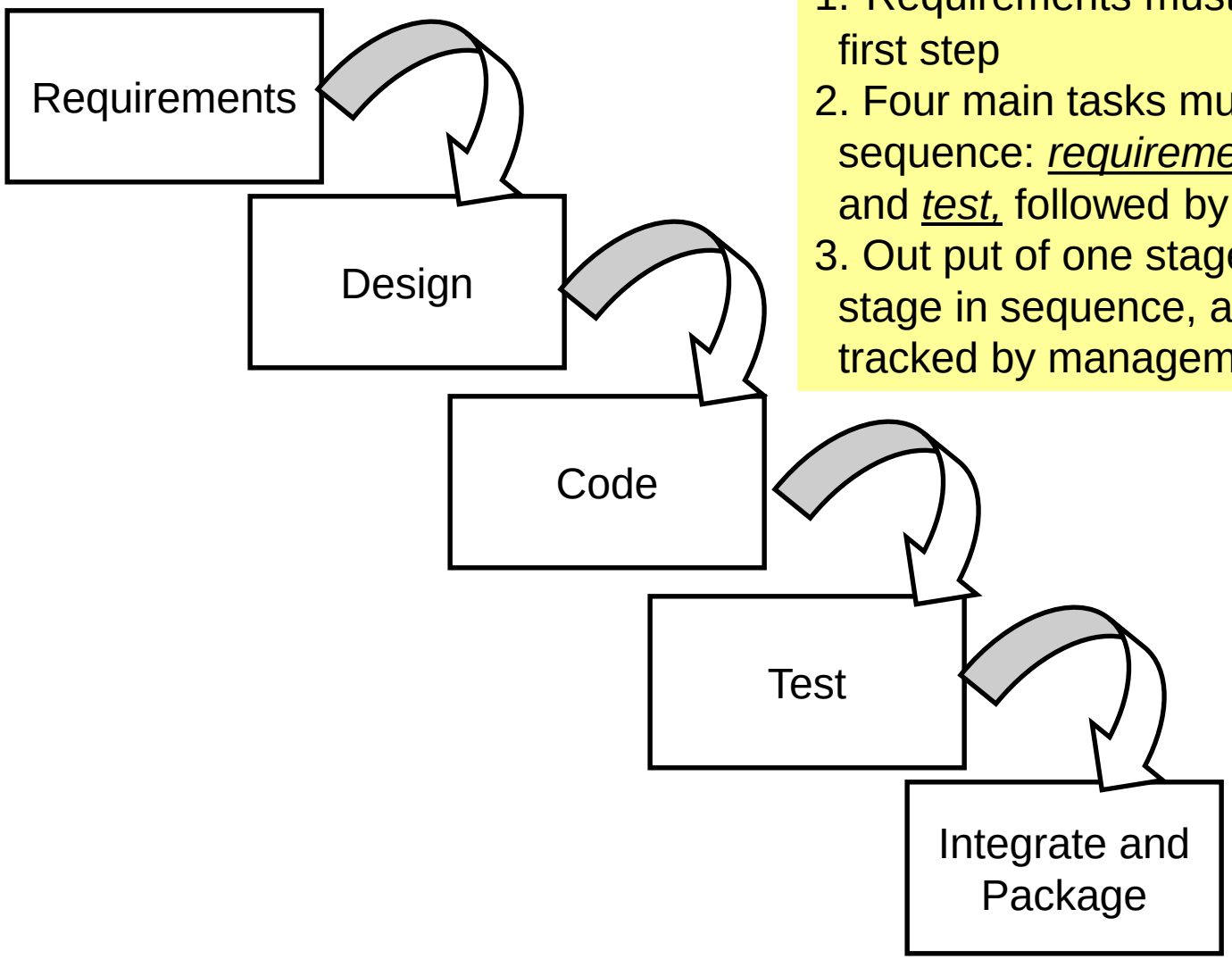


1. Most people performs and follow this process, but unfortunately some skips unit testing or debugging
2. Also, some proceeds without clearly understanding the "problem statement" ---- which is requirement

Some "Traditional" Software Development Processes

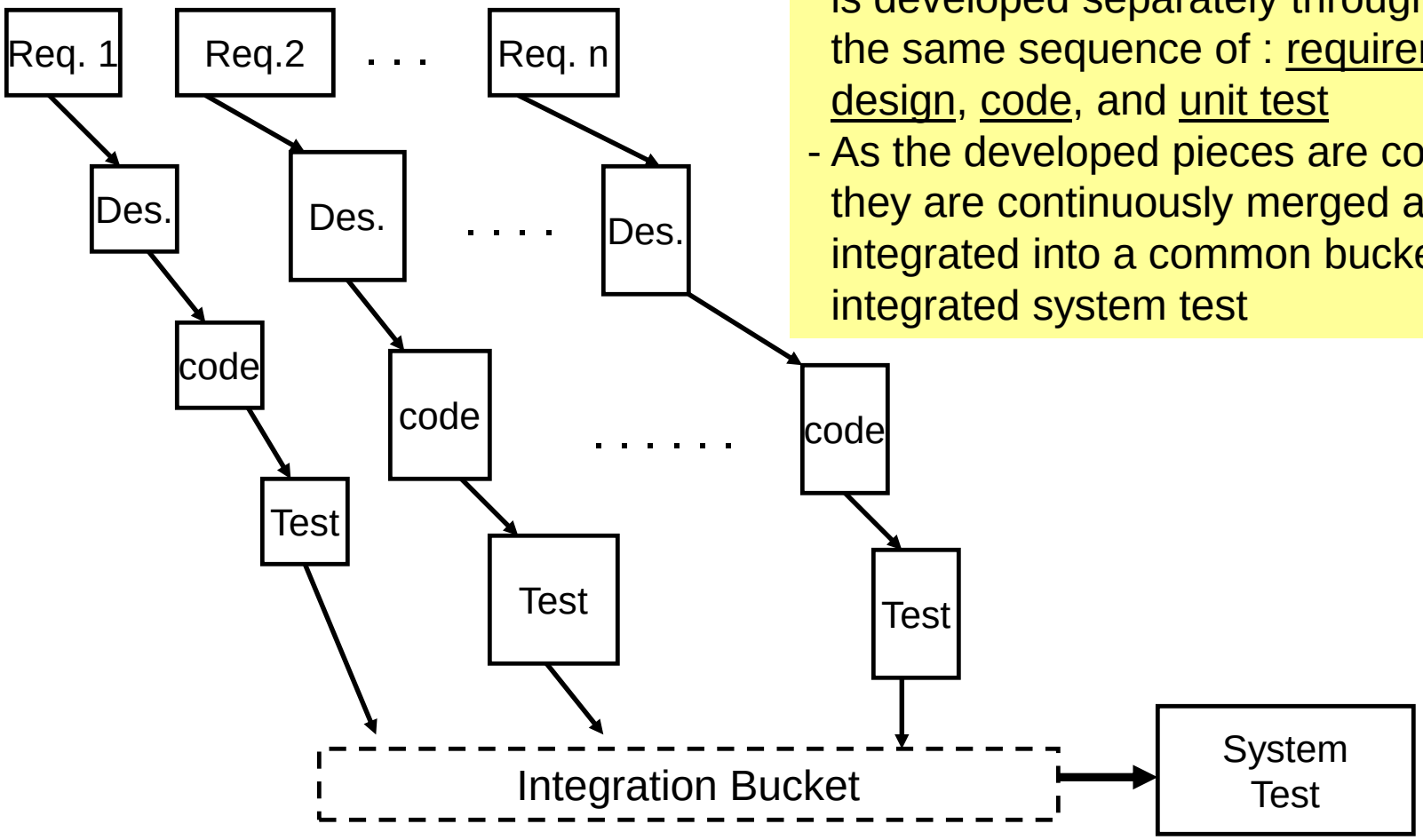
- The "simple" process was employed by many for years without formally embracing other important development activities such as requirements analysis, design, formal testing, or packaging.
- The recognition of the need for formal processes was initially driven by failures in developing large complex software
  - **Waterfall** : earliest process and coping with no process
  - **Incremental** : coping with decomposing the large systems
  - **Spiral** : coping with risk management
  - **Rational Unified Process** : coping with multiple development and management issues

Waterfall Model



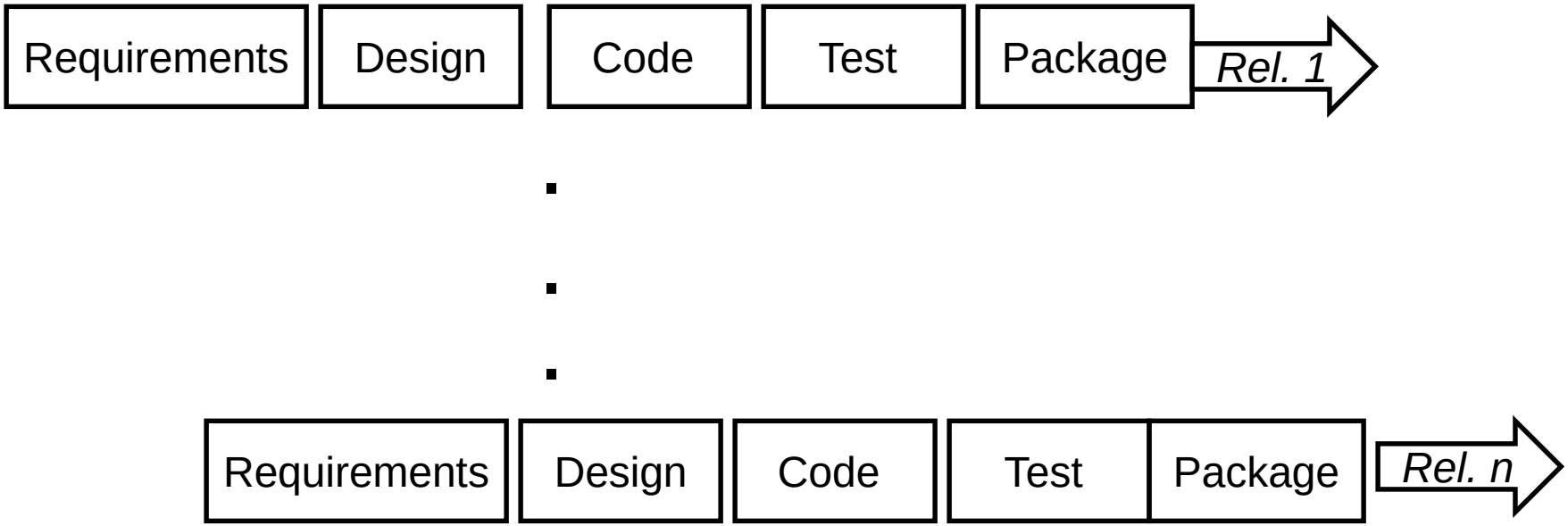
1. Requirements must be specified in the first step
2. Four main tasks must be completed in sequence: *requirements*, *design*, *code*, and *test,* followed by packaging
3. Out put of one stage feeds into the next stage in sequence, and thus easily tracked by management

Incremental Model A – Continuous Integration



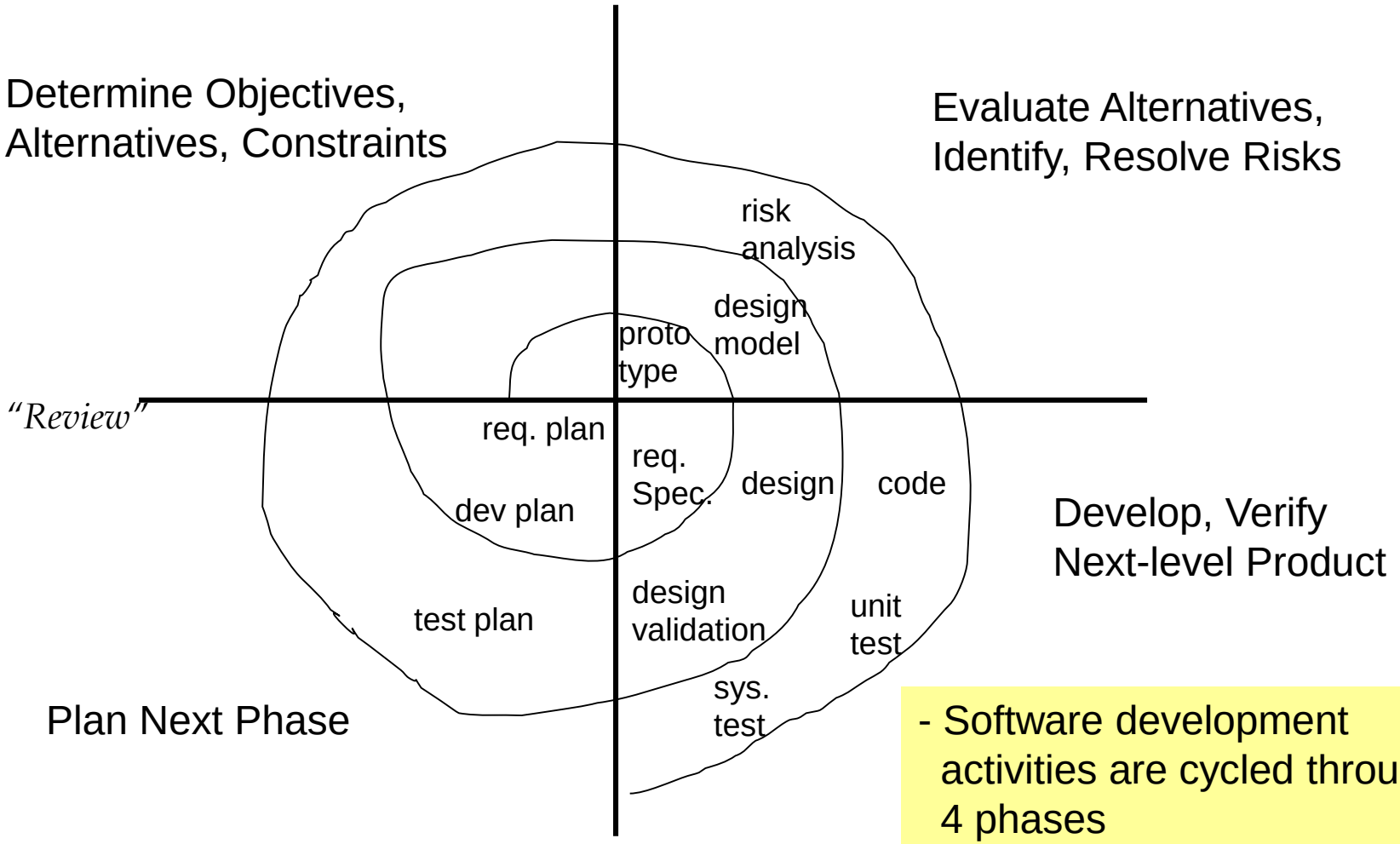
- Each major requirement/item is developed separately through the same sequence of : requirement, design, code, and unit test
- As the developed pieces are completed, they are continuously merged and integrated into a common bucket for integrated system test

Incremental Model B - Multiple Release



Each small set of requirements is developed, packages, and released in a multiple release Fashion

Spiral Model



- Software development activities are cycled through 4 phases
- A Risk averse process

Problems with Traditional Processes

Lengthy development time

Inability to cope with changes in requirements

Assumes requirements are understood at beginning of project

Relies on heroic development effort

Complex methodology

Waste/duplication of effort

Software Process in Modern Software Development: Introduction to Agile

Agile vs. Waterfall: A Tale of Two Teams

The key characteristics of Agile Process

- Family of software development methodologies
- Short releases and iterations
- Incremental design
- User involvement
- Minimal documentation
- Informal communications
- Change

The Manifesto for Agile Software Development

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

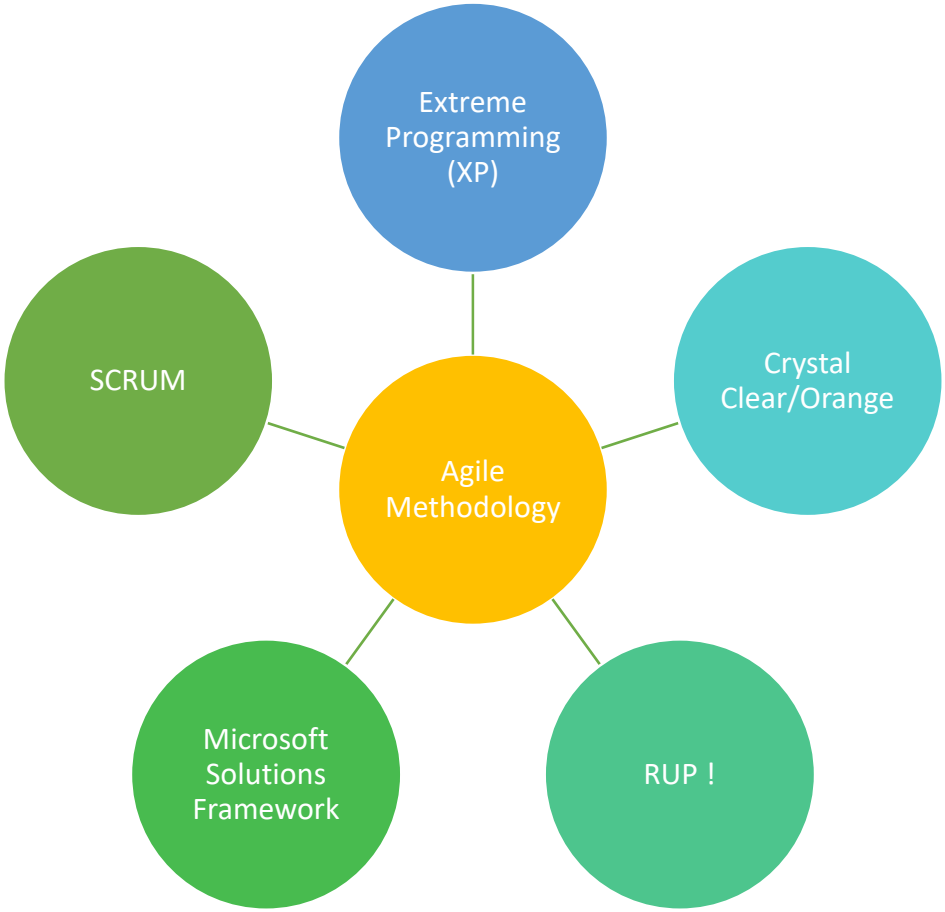
- *Individuals and interactions* over processes and tools
- *Working software* over comprehensive documentation
- *Customer collaboration* over contract negotiation
- *Responding to change* over following a plan

That is, while there is value in the items on the right, we value the items on the left more."

http://agilemanifesto.org

Kent Beck et al

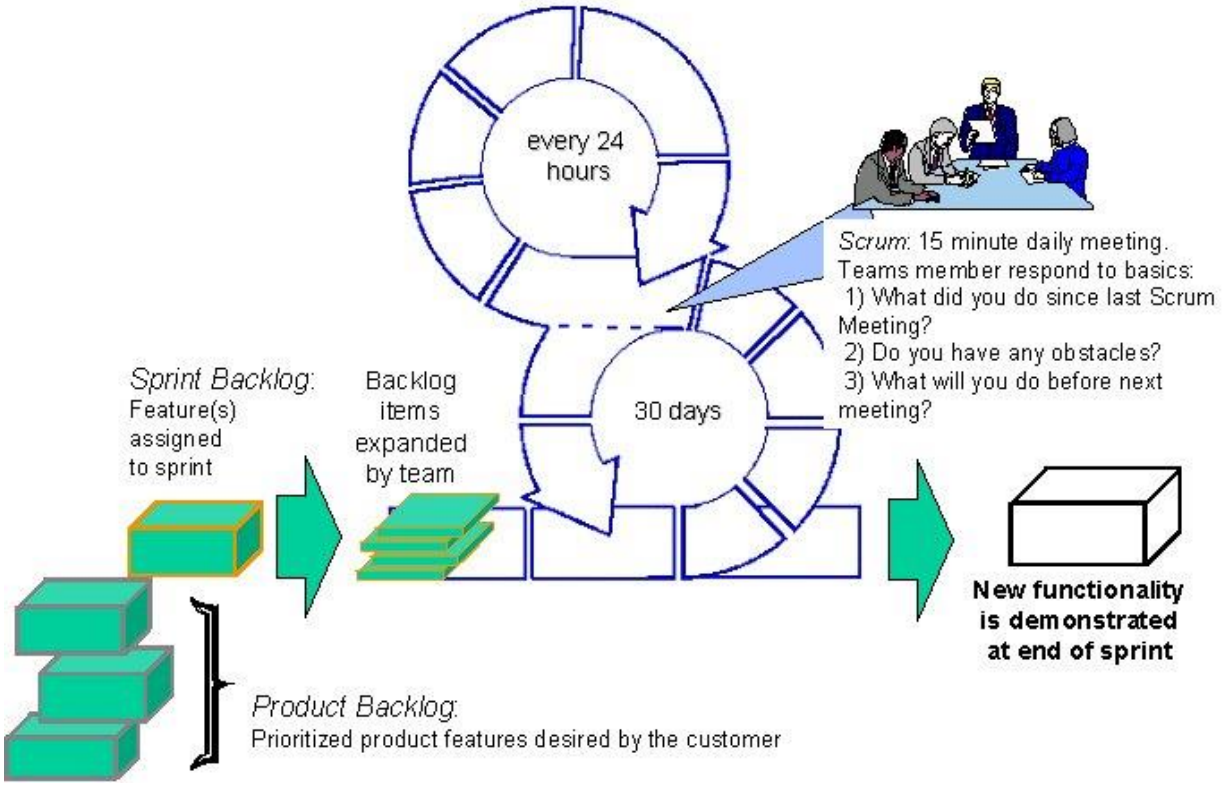
Some Agile Methodologies



Scrum

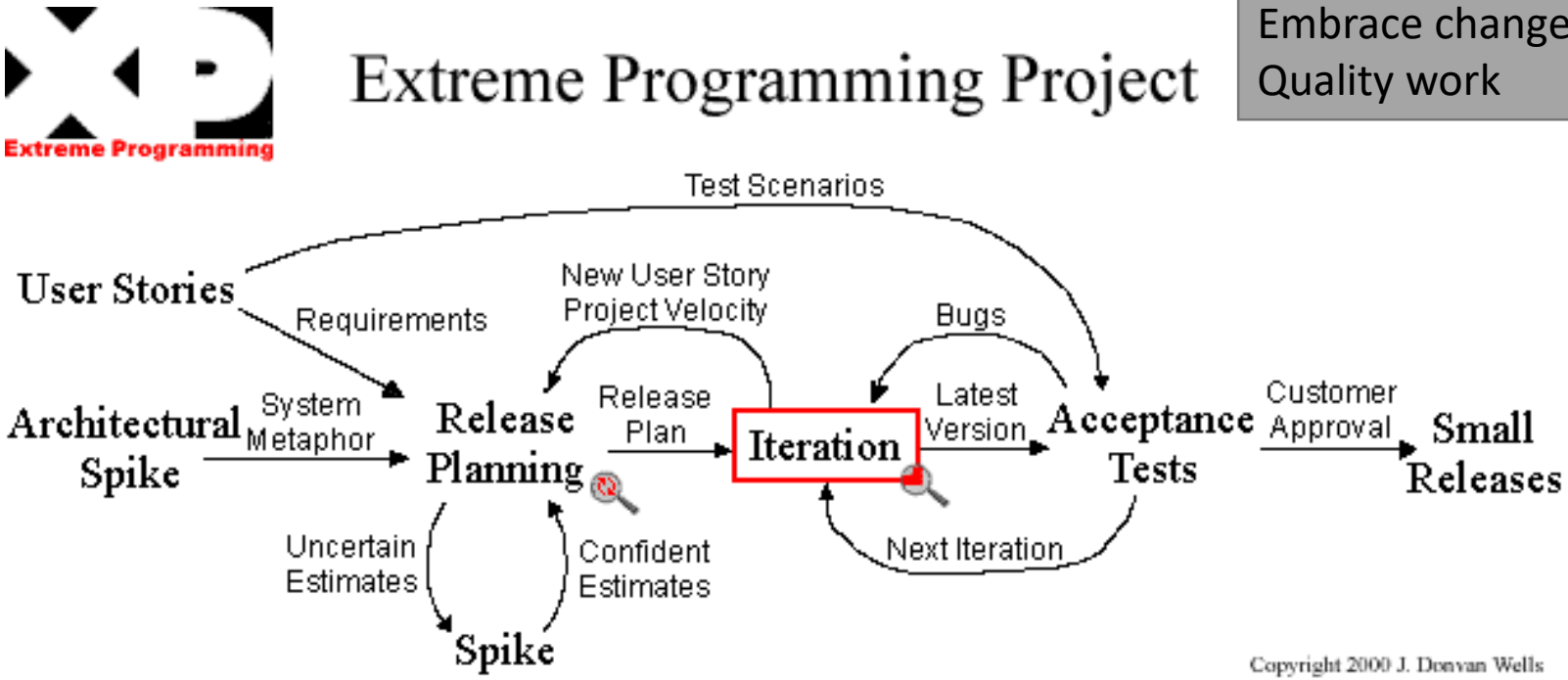
- Originally proposed by Schwaber and Beedle
- Scrum—distinguishing features
  - Development work is partitioned into "packets"
  - Testing and documentation are on-going as the product is constructed
  - Work occurs in "sprints" and is derived from a "backlog" of existing requirements
  - Meetings are very short and sometimes conducted without chairs
  - "demos" are delivered to the customer with the time-box allocated

Scrum (cont.)


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

Extreme Programming (XP)

- Rapid feedback
- Simplicity
- Incremental change
- Embrace change
- Quality work

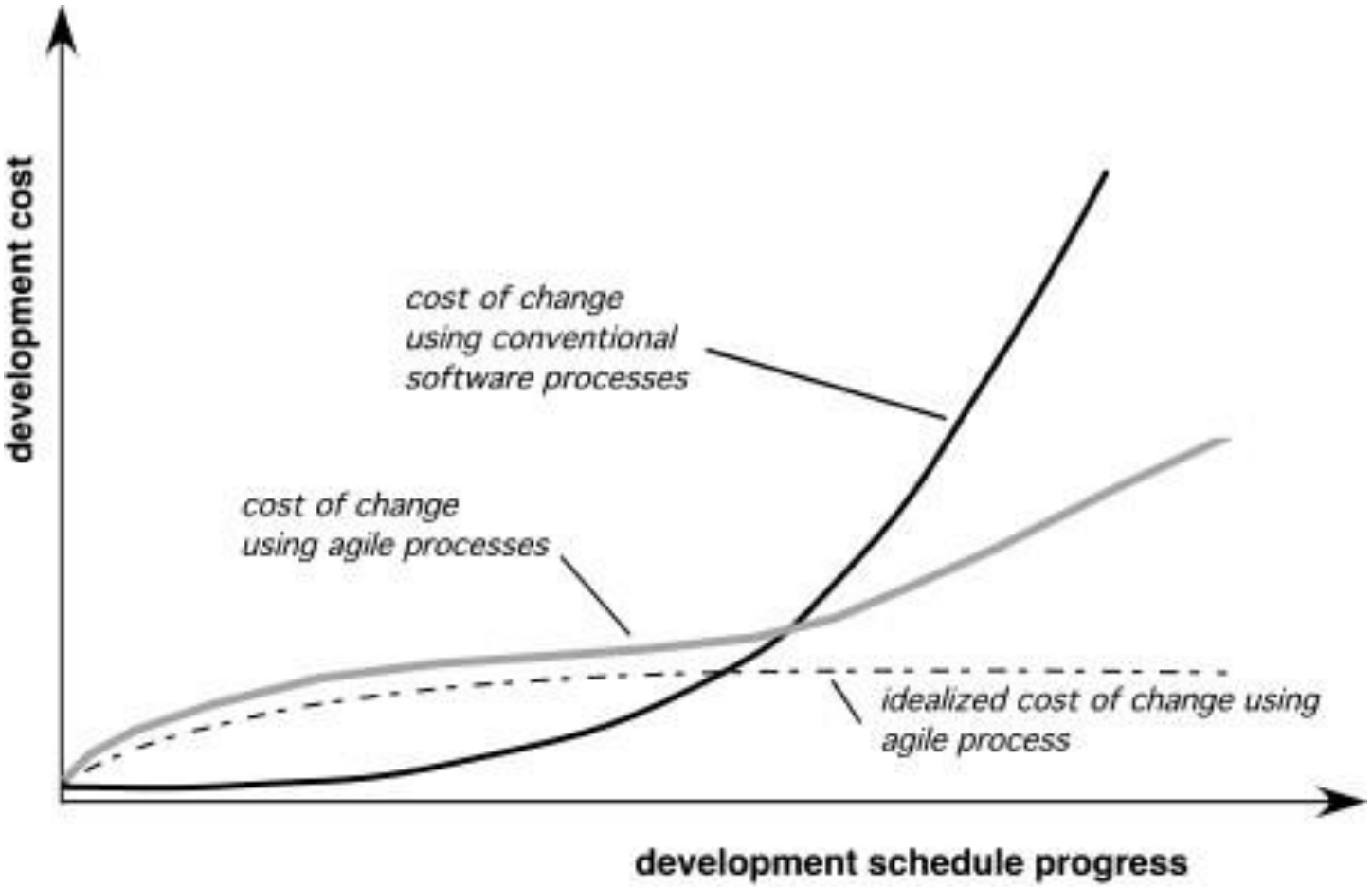


Source: http://www.extremeprogramming.org/

Agile vs. Traditional

	Agile	Traditional / Heavy
Requirements	• Assumes change • Informal requirements • Constant user interaction	• Assumes no change • Complete, detailed, formal requirements document
Design	• Informal • Iterative	• Formal • Upfront
User involvement	• Crucial • Frequent	• Beginning (Requirements) • End (Acceptance testing)
Documentation	• Minimal, only as needed • Source code	• heavy, formal documents
Communication	• Informally • Throughout the project	• Documents • Formal memos and meetings
Complexity	• Low	• High
Overhead	• Low	• High

Agility and the Cost of Change



Agile vs. Traditional

Advantages

- Simpler
- Low cost, overhead
- Deals with changes
- Fast results
- Usable systems

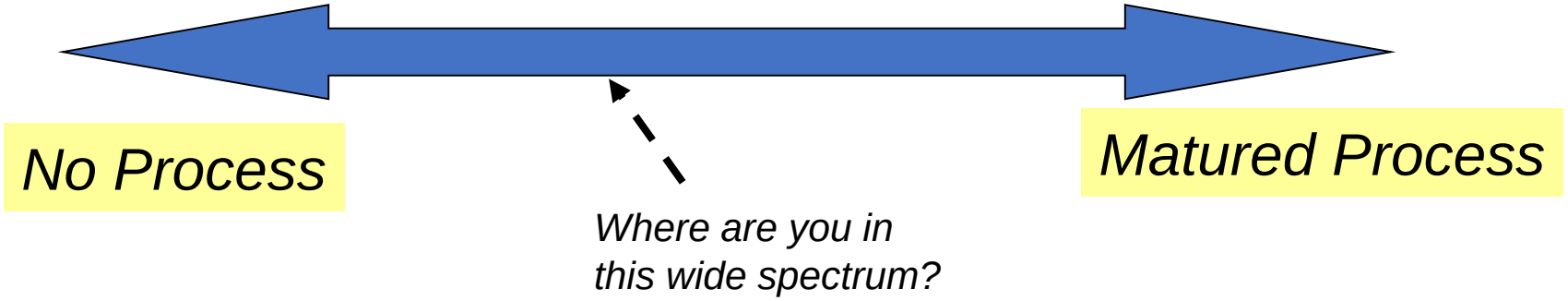
Risks, Disadvantages

- Not scalable
- Relies on teamwork
- Relies on access to customer
- Cultural clash

Software Process

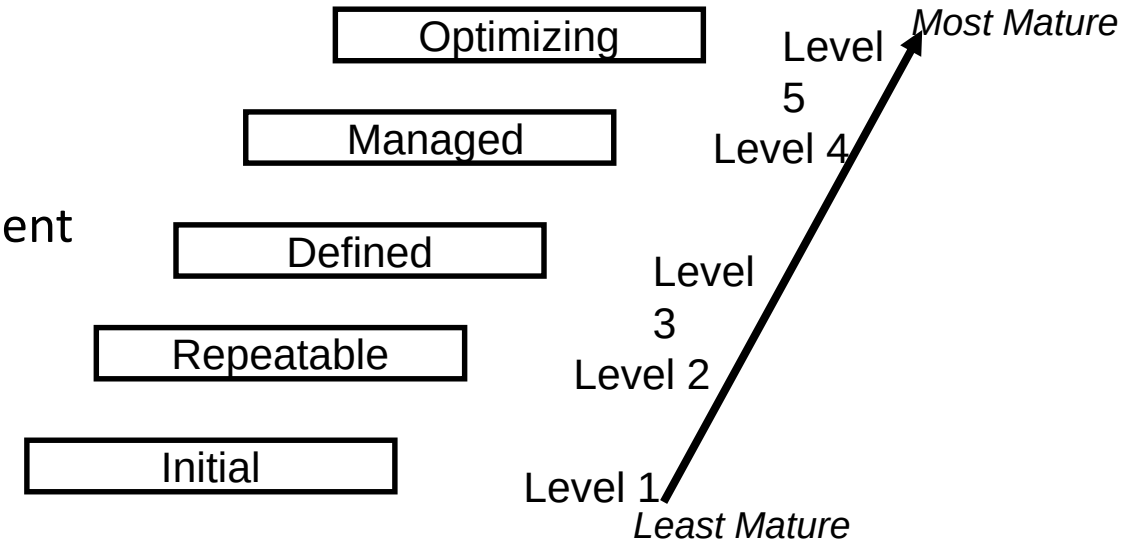
Process Assessment

- How mature is your software engineering organization and do you need to improve?
- ISO (ISO 9000 series) and SEI (at Carnegie Mellon) are two leading organizations that help in the *process assessment*

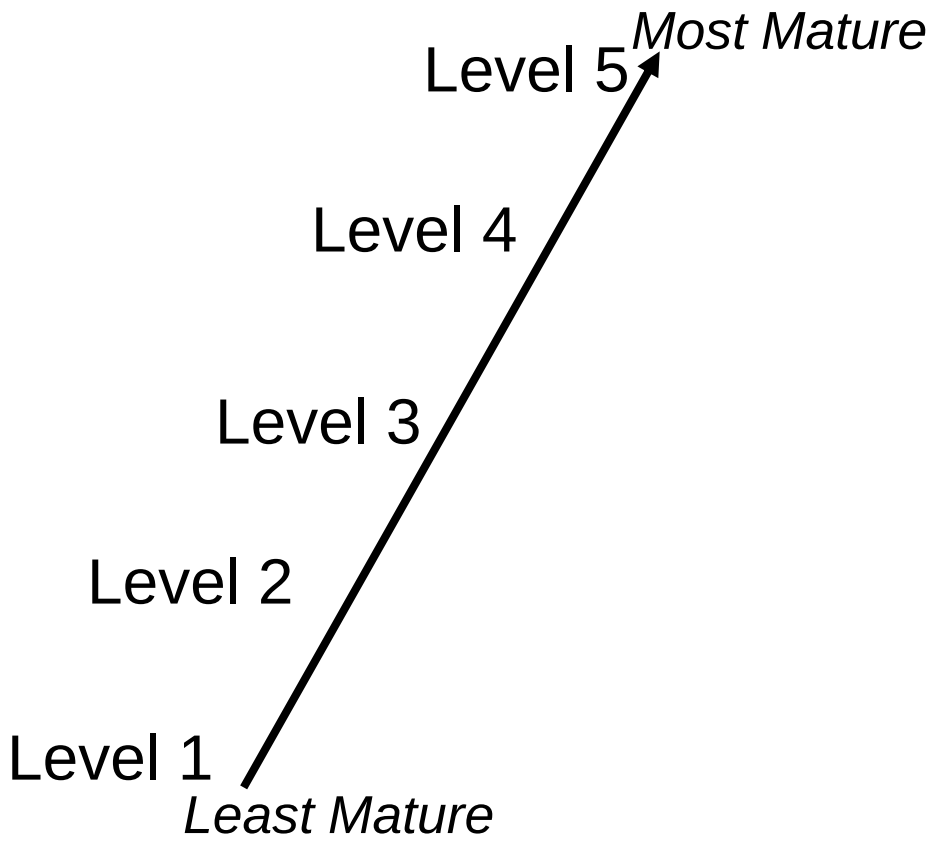


SEI CMM & CMMI

- Software Engineering Institute (SEI) proposed a Capability Maturity Model (CMM) to help software organizations assess their maturity and provide guidance in software development
- In 2001, CMM was upgraded to CMMI (CMM Integrated). There are multiple major aspects to CMMI:
  - Systems engineering
  - Software engineering
  - Integrated product and process development
  - Supplier sourcing



Example of SPI: CMMI Level 1



Initial

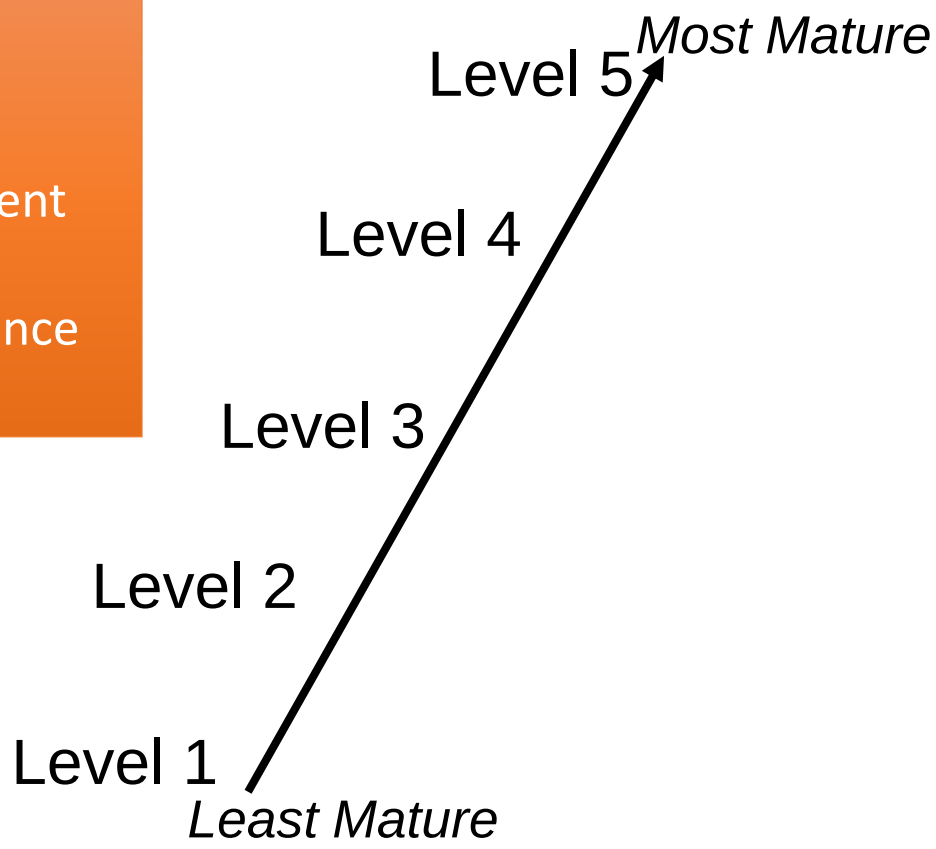
Ad hoc and Chaotic

Example of SPI: CMMI Level 2

1) Requirements Management
2) Project Planning
3) Project Monitoring and Control
4) Supplier Agreement and Management
5) Measurement and Analysis
6) Process and Product Quality Assurance
7) Configuration Management

Repeatable

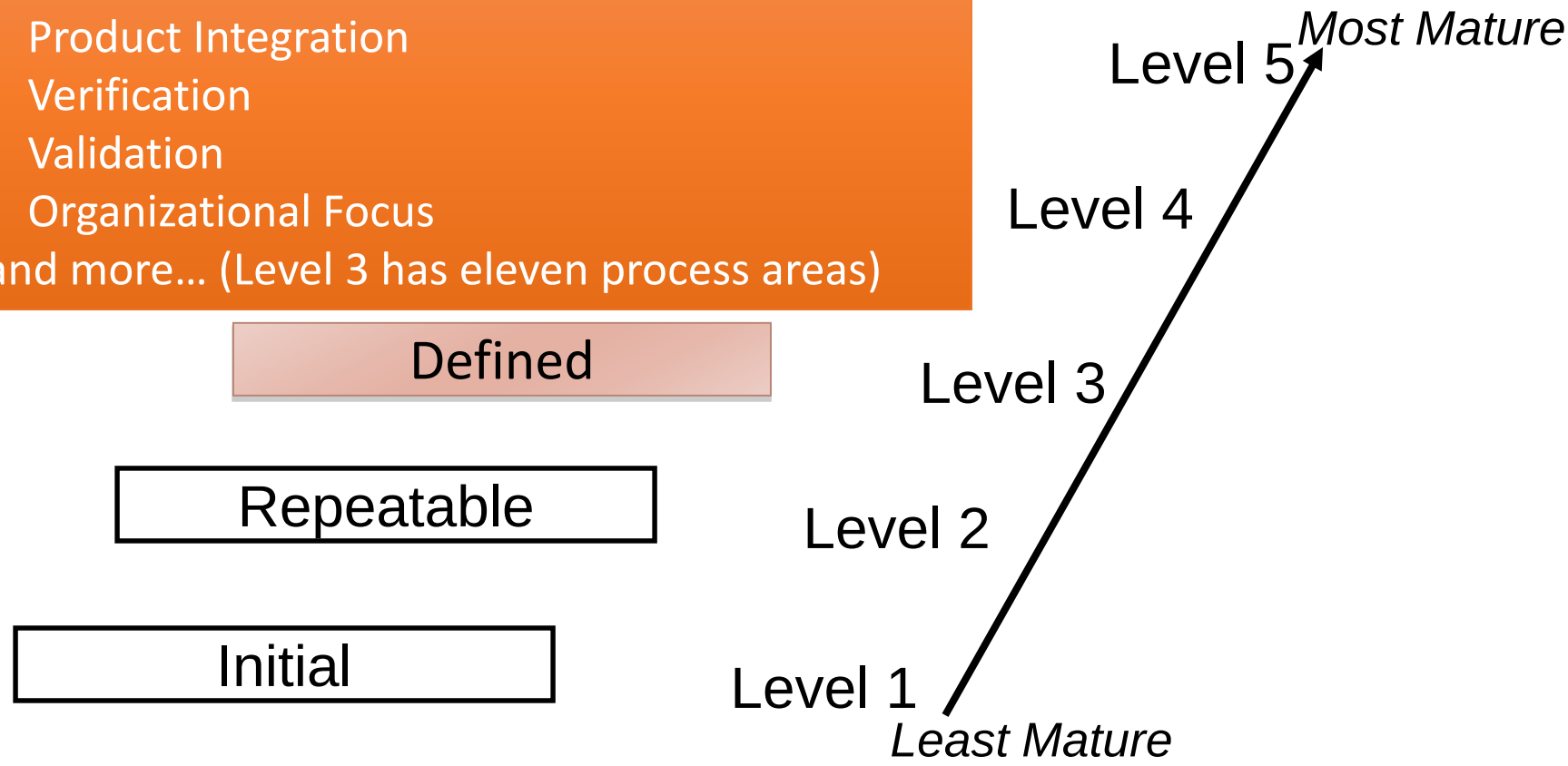
Initial



Example of SPI: CMMI Level 3

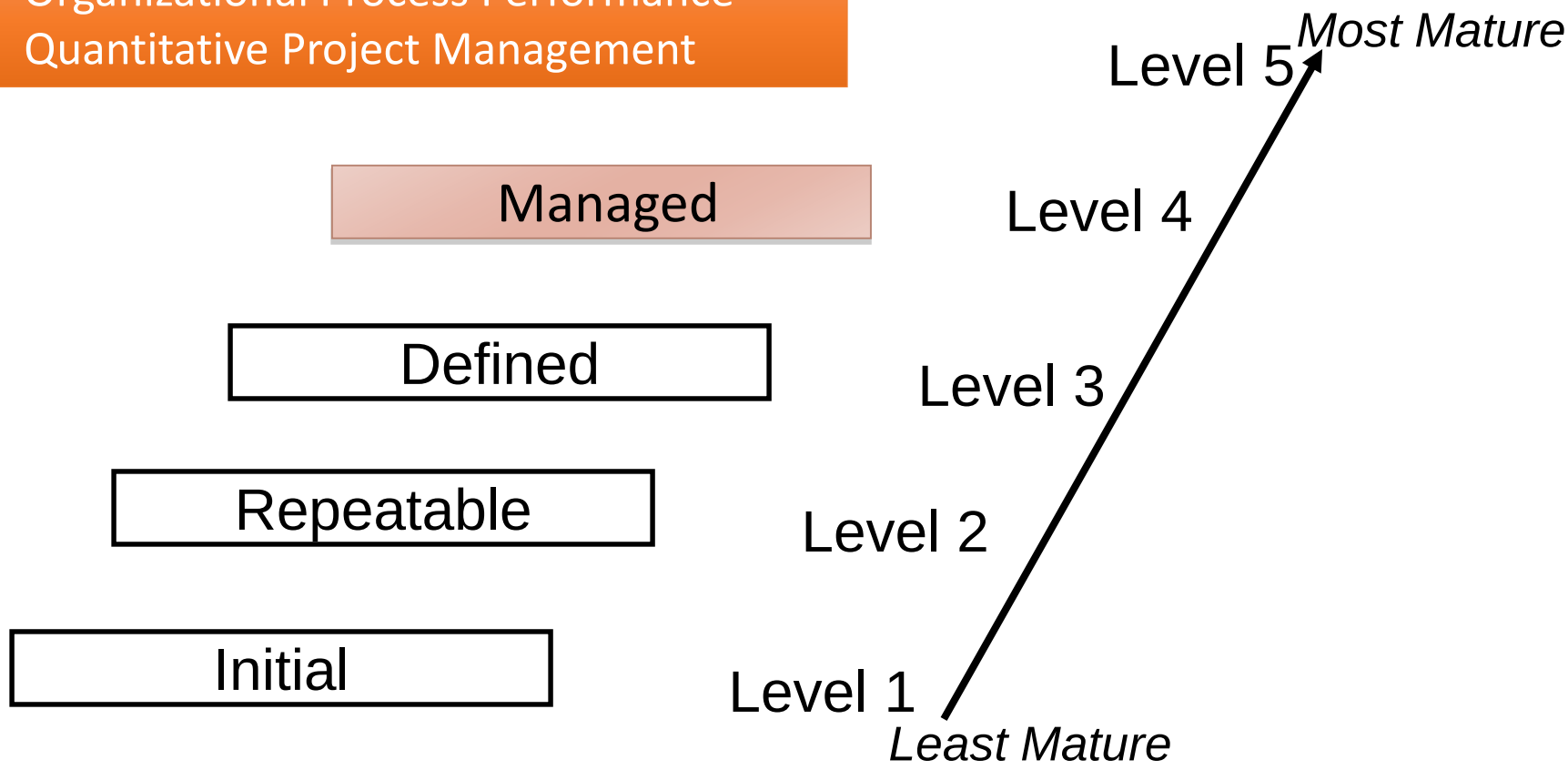
1) Requirements Development
2) Technical Solution
3) Product Integration
4) Verification
5) Validation
6) Organizational Focus

… and more… (Level 3 has eleven process areas)



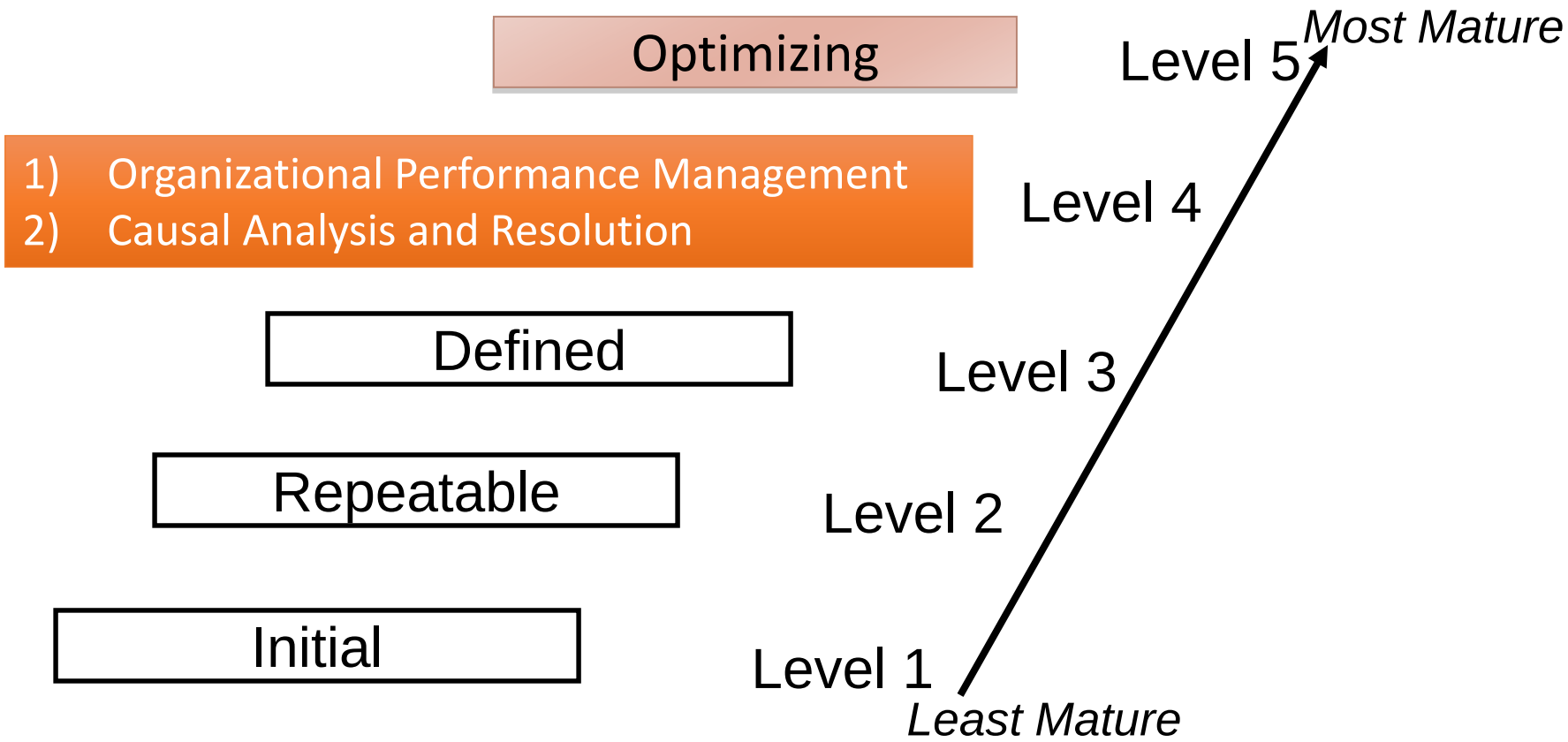
Example of SPI: CMMI Level 4

1) Organizational Process Performance
2) Quantitative Project Management



Example of SPI: CMMI Level 5

Goal: Better Process -> Better Project Management -> Better Product



Software Project Management

Software Project Management

AN UMBRELLA ACTIVITY WITHIN SOFTWARE ENGINEERING

BEGINS BEFORE ANY TECHNICAL ACTIVITY AND CONTINUE THROUGHOUT SOFTWARE DEVELOPMENT PROCESS

The Management Spectrum (Four P's)

Problem

- ***Objectives***
- ***Scope***
- ***Alternatives***
- ***Technical and management constraints***

People

- ***Roles***

Project

Process

- ***Tasks***
- ***Milestones***
- ***Deliverables***
- ***QA***

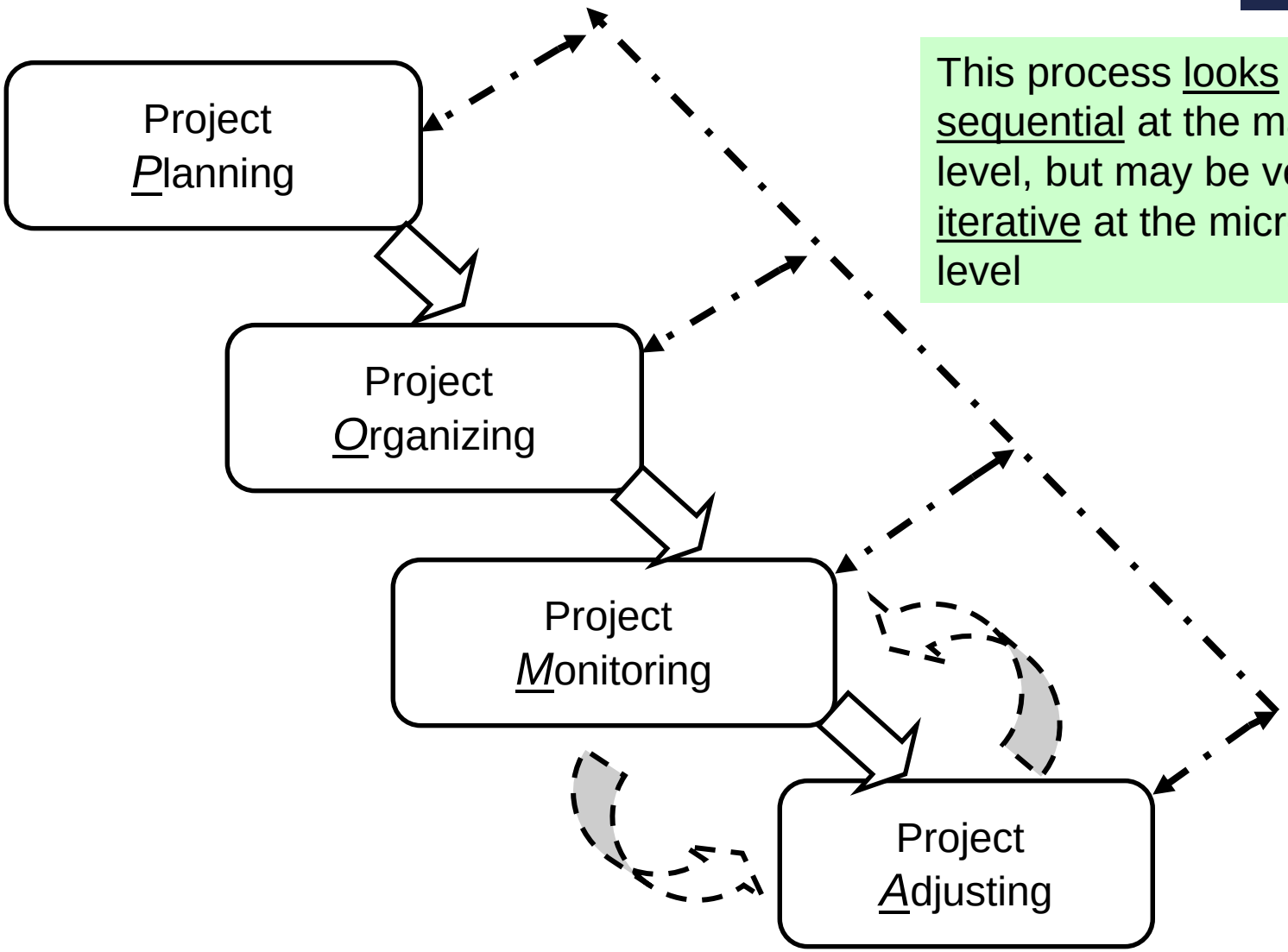
Project Management Process

- Why do we need project management?
- Why can't we just follow one of the software development process and be left alone?

All projects – small and large – need project management because all projects need some degree of

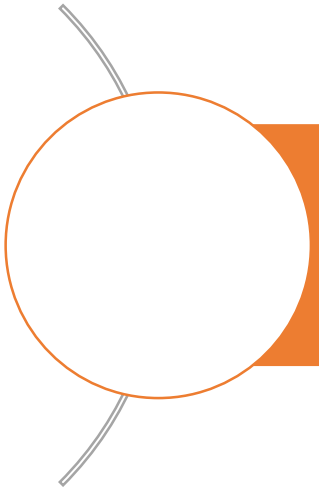
- planning
- organizing
- status monitoring
- adjustment

Software Project Management (POMA) Process



This process looks sequential at the macro level, but may be very iterative at the micro level

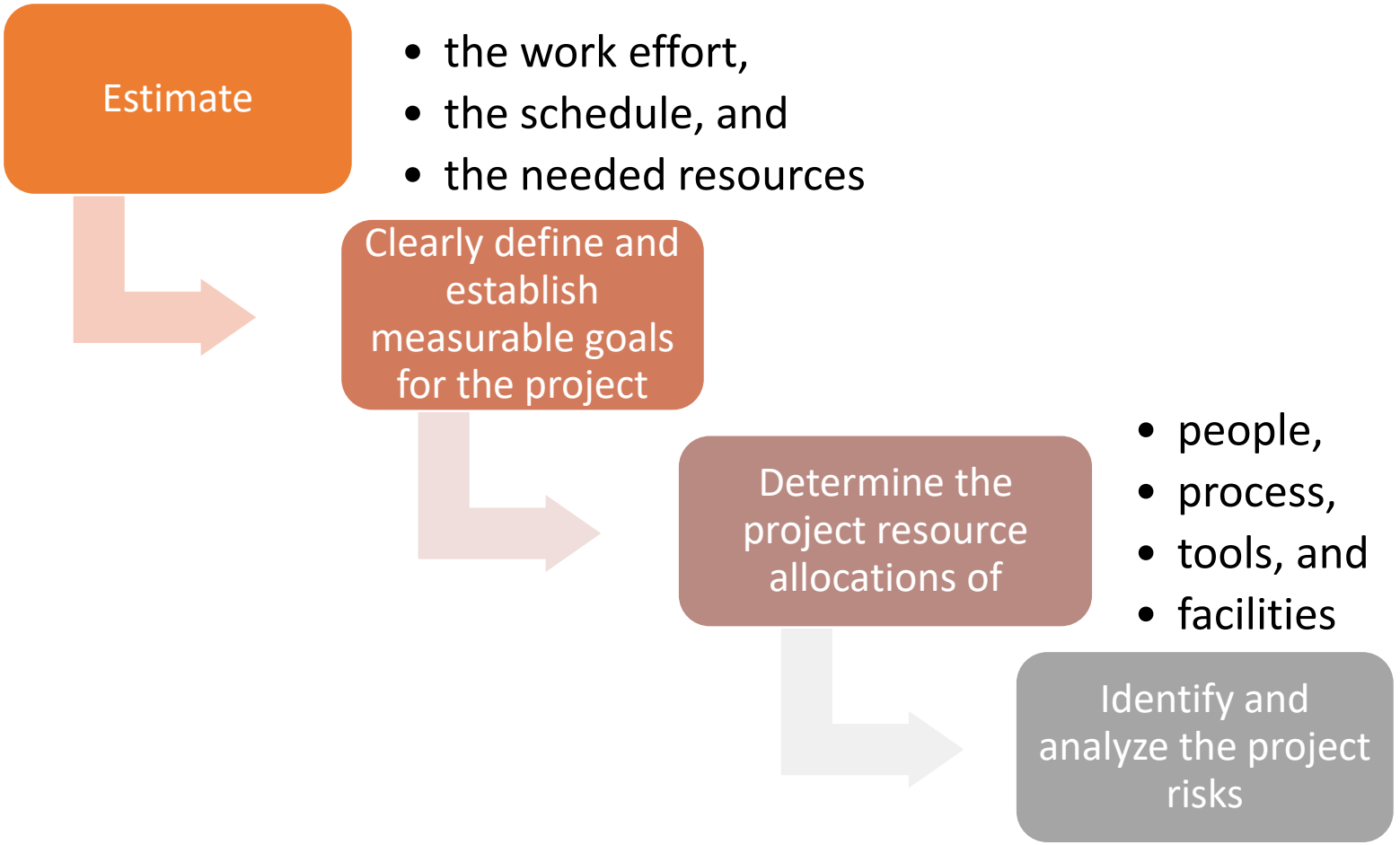
Planning (POMA)



The 1st step of project planning is to understand the requirements of the project. This step itself may be a mini-project

Planning (POMA) (cont.)

Then the following 4 steps are included in the rest of project planning



Organizing (POMA)

Once a project plan is formulated or partially formulated, organizing may start

Organization structure needs to be designed

Human resource hiring needs to start and be completed along with acquisition of other resources

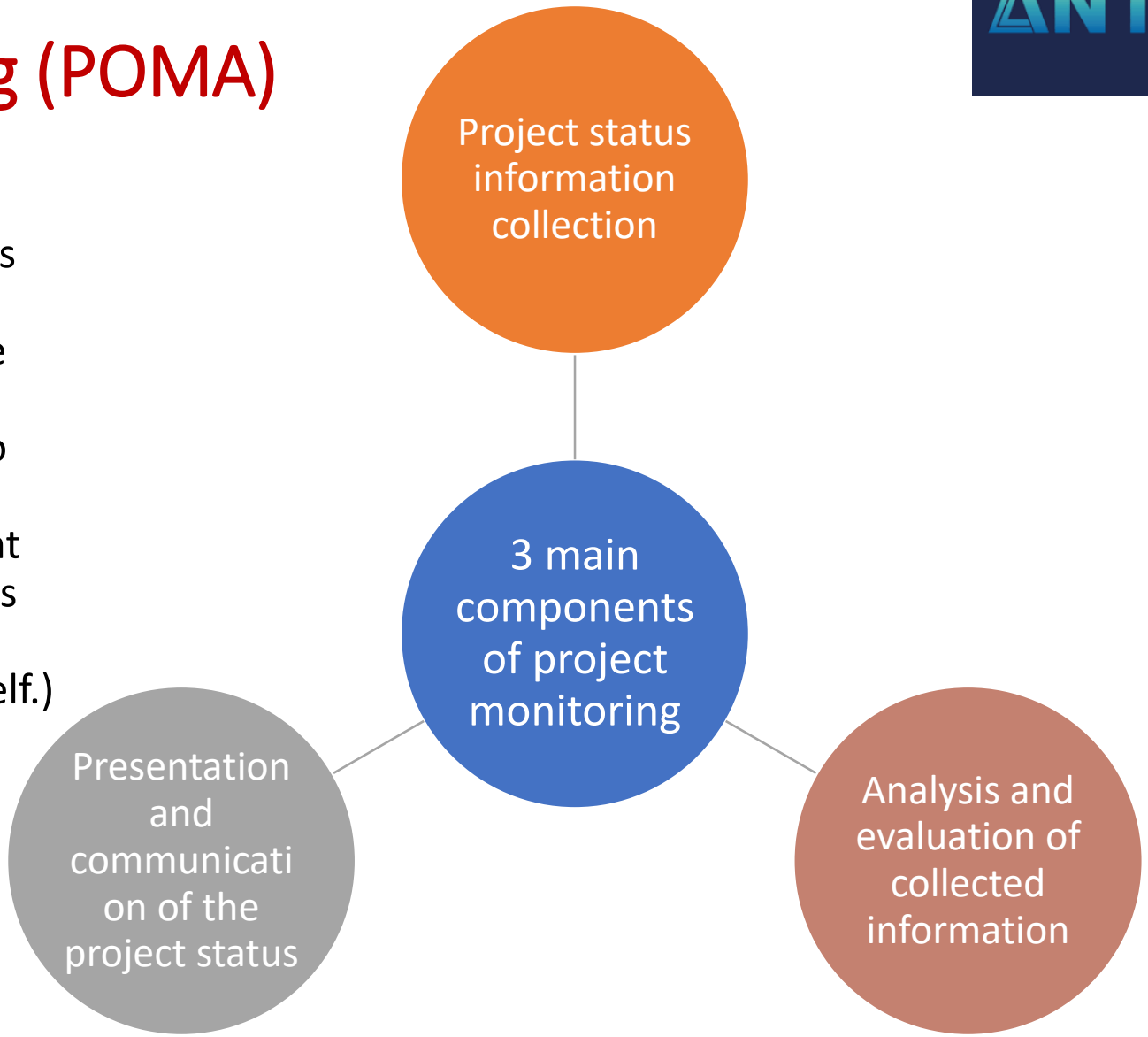
Any required education and training have to be completed

Mechanisms for tracking must be established

- Risk tracking and mitigation
- Project goal monitoring

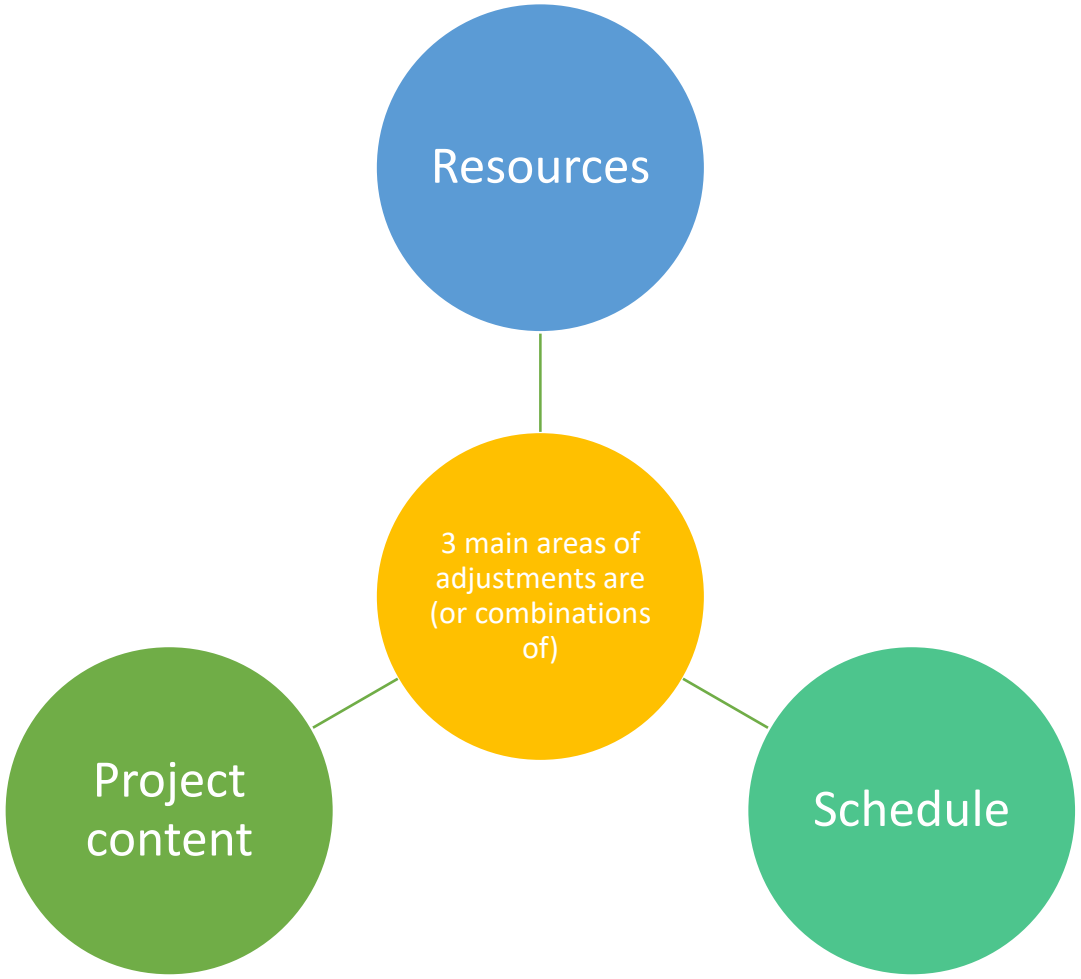
Monitoring (POMA)

Once the project is organized and set into motion, there still needs to be regular tracking to ensure that it is headed in the right direction. (Projects can not be left to coast along by itself.)



Adjusting (POMA)

It is highly unlikely that a software project progresses with no problem. As soon as the project status suggests potential problem, we must not be afraid to make changes.



Project Management Framework

Stakeholders' needs and expectations

9 Knowledge Areas

Core Functions

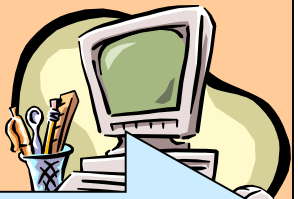
Scope Mgt | Time Mgt | Cost Mgt | Quality Mgt

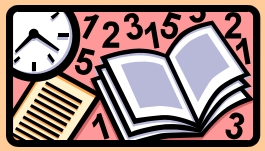
Project Management Integration

HR Mgt | Comm. Mgt | Risk Mgt | Procure. Mgt

Facilitating Functions

Tools and techniques





Project Success

PMBOK, 1996

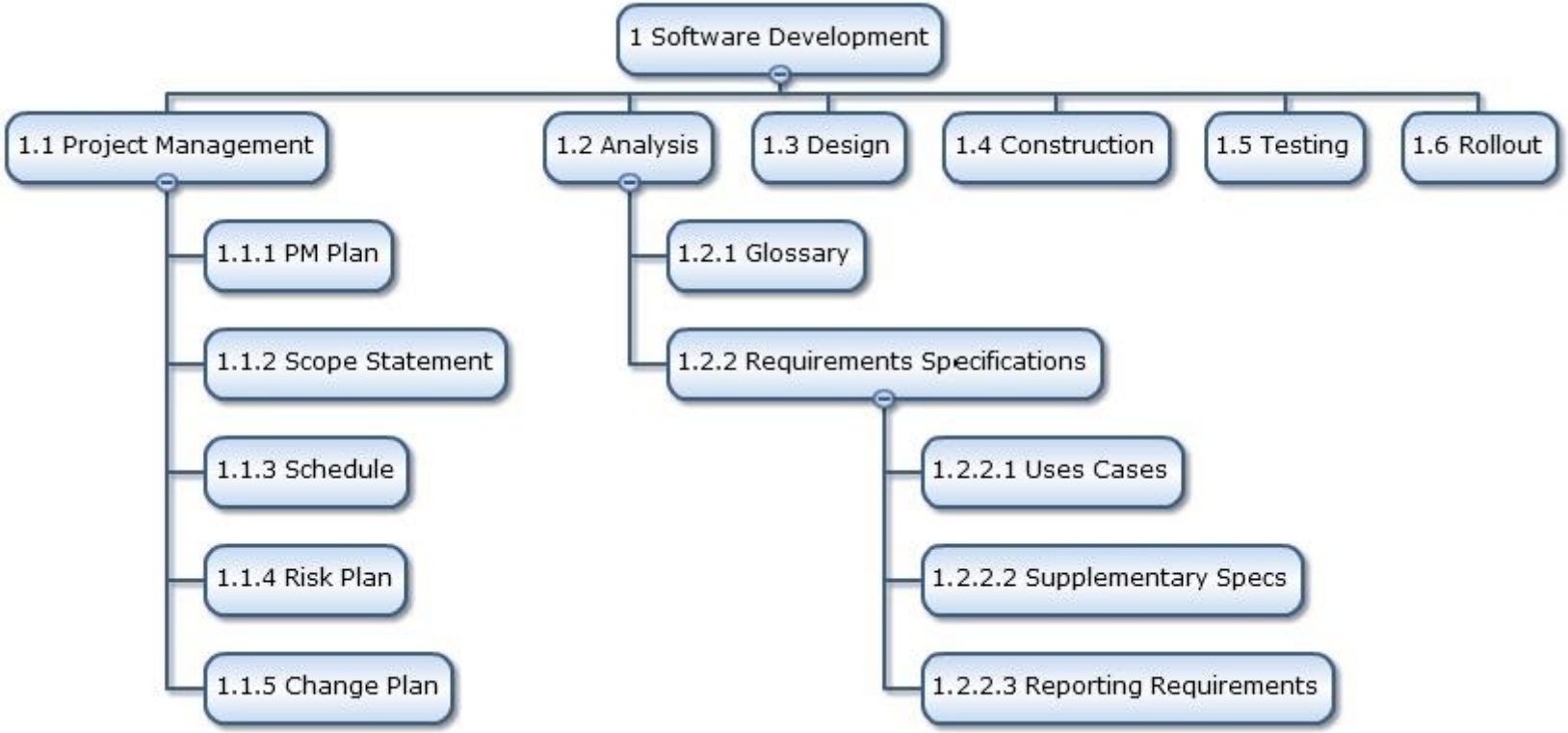
Software Project Management Process

is not the same as

- Software Development Process or
- Software Life Cycle

Start Planning Your Project…

- Know your project goal, scope and deliverables



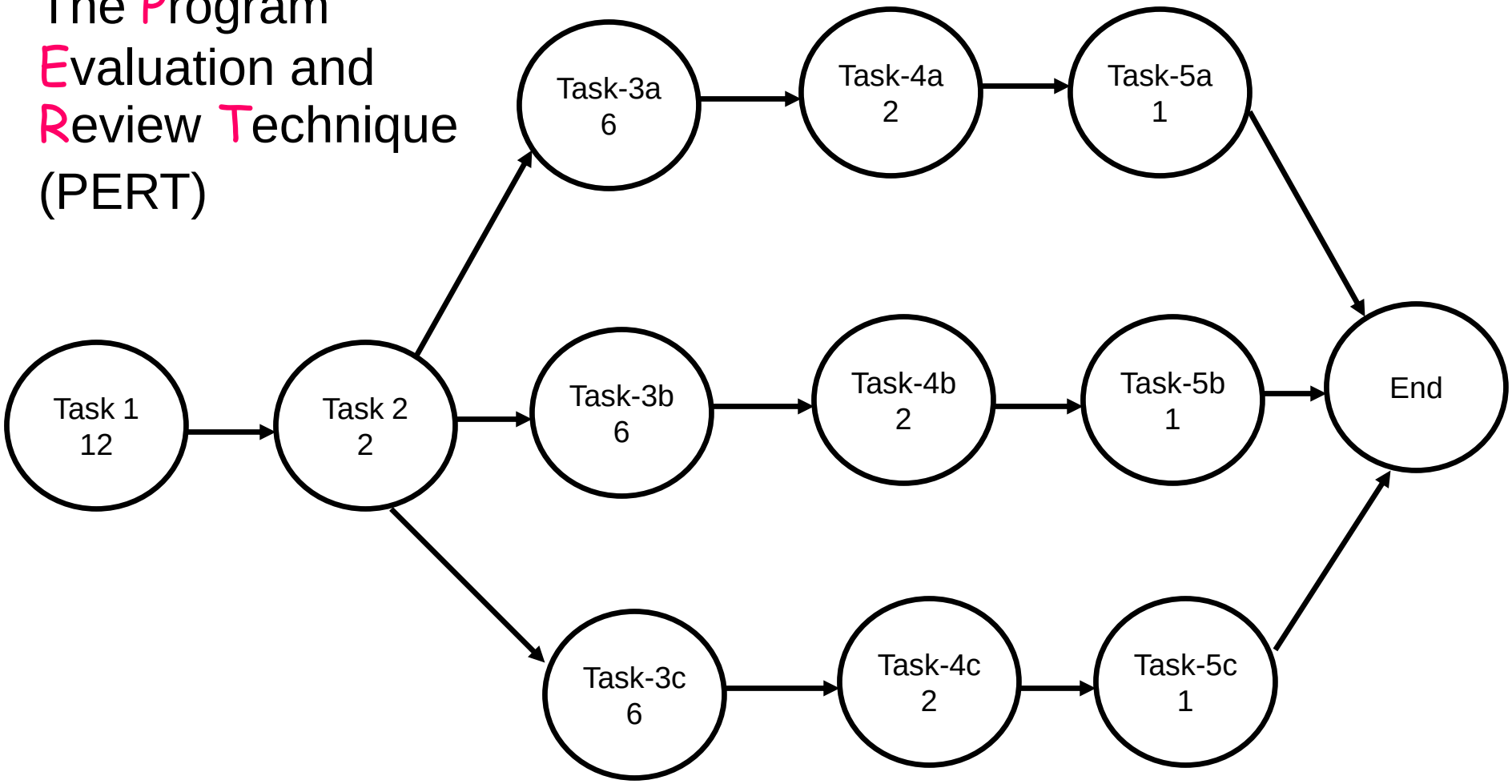
Then…

Use WBS to

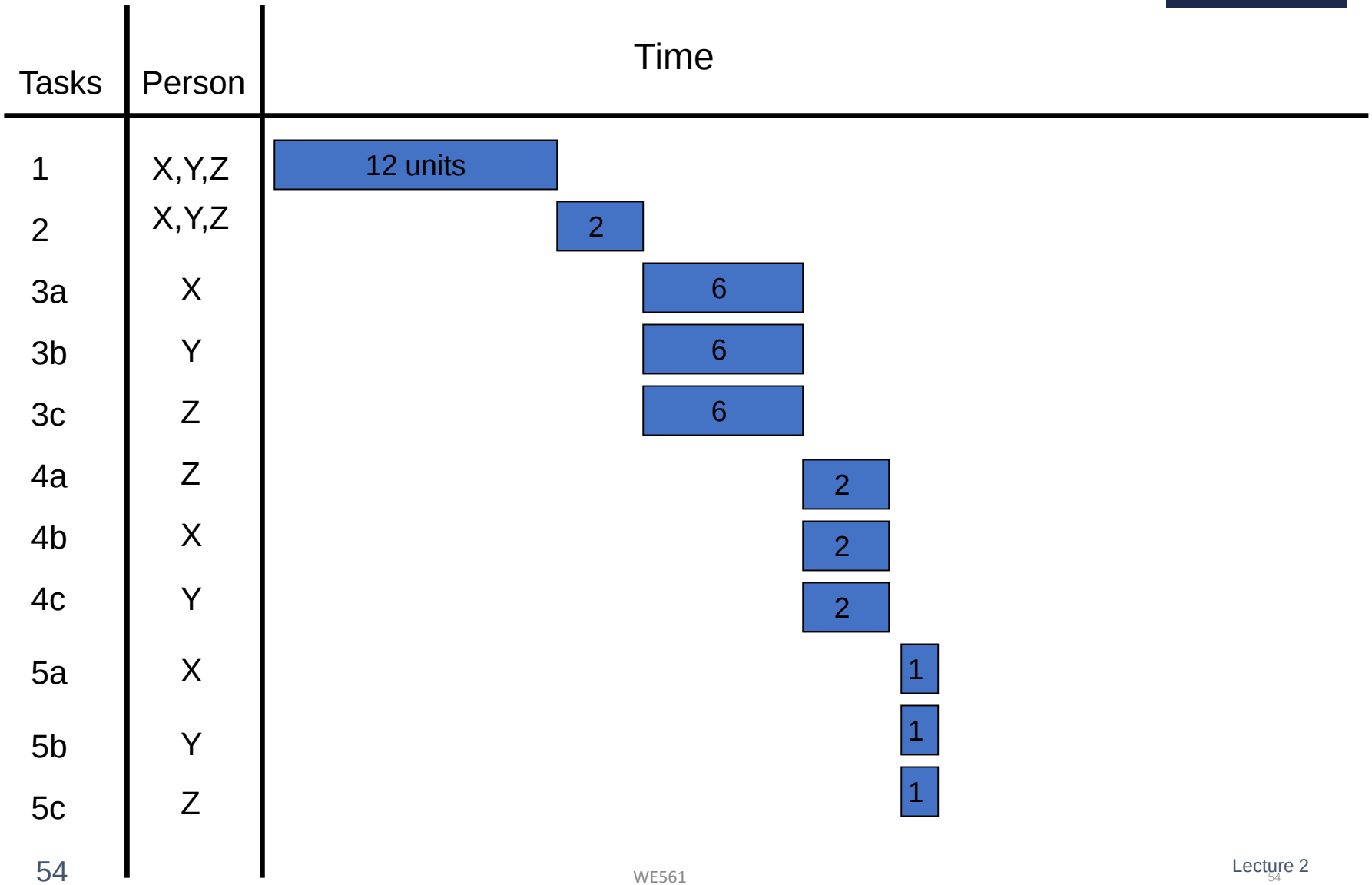
- Examine and determine the external deliverables of the project
- Identify the steps and tasks required to produce each of the deliverables, including the tasks that are required to produce any intermediate internal deliverables
- Sequence the tasks, showing any potential for parallelism
- Provide an estimate to complete each of the tasks
- Provide an estimate of the productivity of the personnel that is most likely to be assigned to each of the tasks
- Calculate the time required to accomplish each task
- For each of the external deliverable, lay out the timeline of all the tasks needed to produce that deliverable and label the resources that will be assigned to the tasks

Sample Technique for Time and Resource Planning (1)

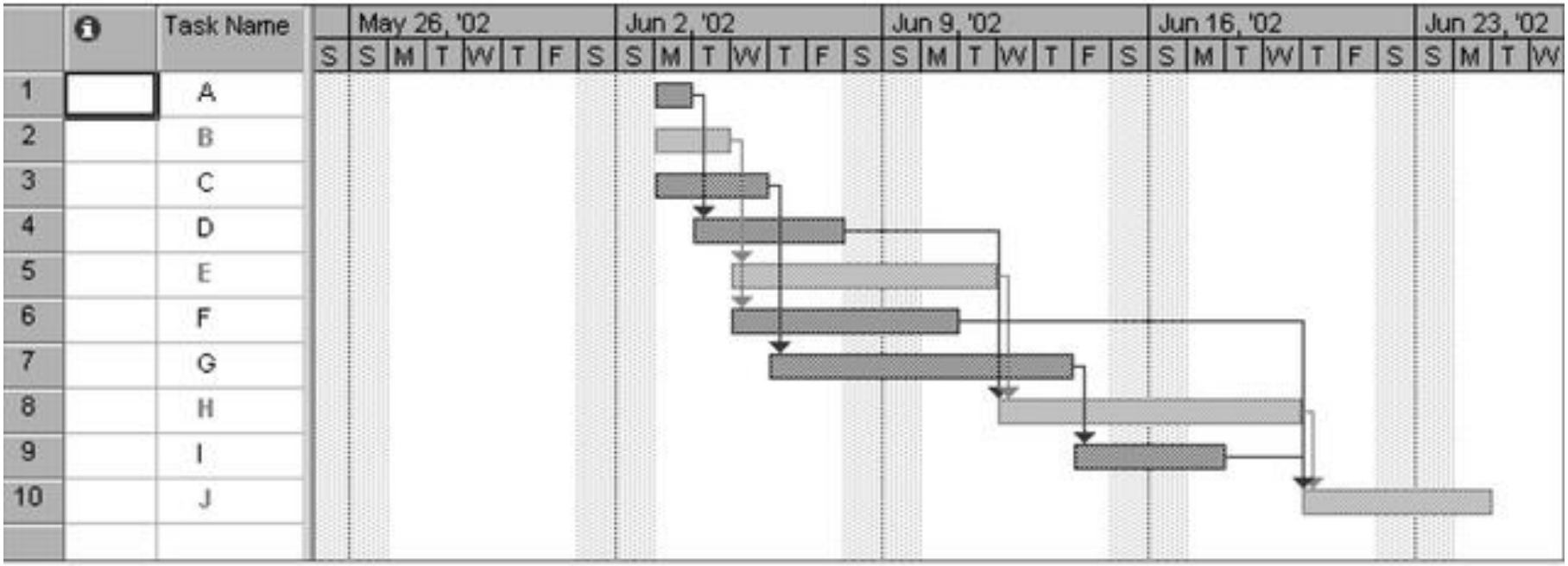
The Program Evaluation and Review Technique (PERT)



End Result of WBS = Initial Schedule Estimate



Gantt Chart for Project X



Goals of Software Project Management

End results of the project satisfy the customer's needs
All the desired and the needed product / project attributes (quality, security, productivity, cost, etc.) are met
Target milestones and the overall schedule are met
Team members are operating effectively and at a high level of morale
Required tools and other resources are made available and are effectively utilized

Summary

Software process describes approaches to a variety of tasks or activities that take place to development software while project management process is a set of tasks or activities that take place to manage the development

Software project management process requires different set of skills from software development process

Failure in project management cause more problems than failure in the technology

Classwork

- Create product plan for your term project
- Schedule your plan by using MS Project/ any tools