

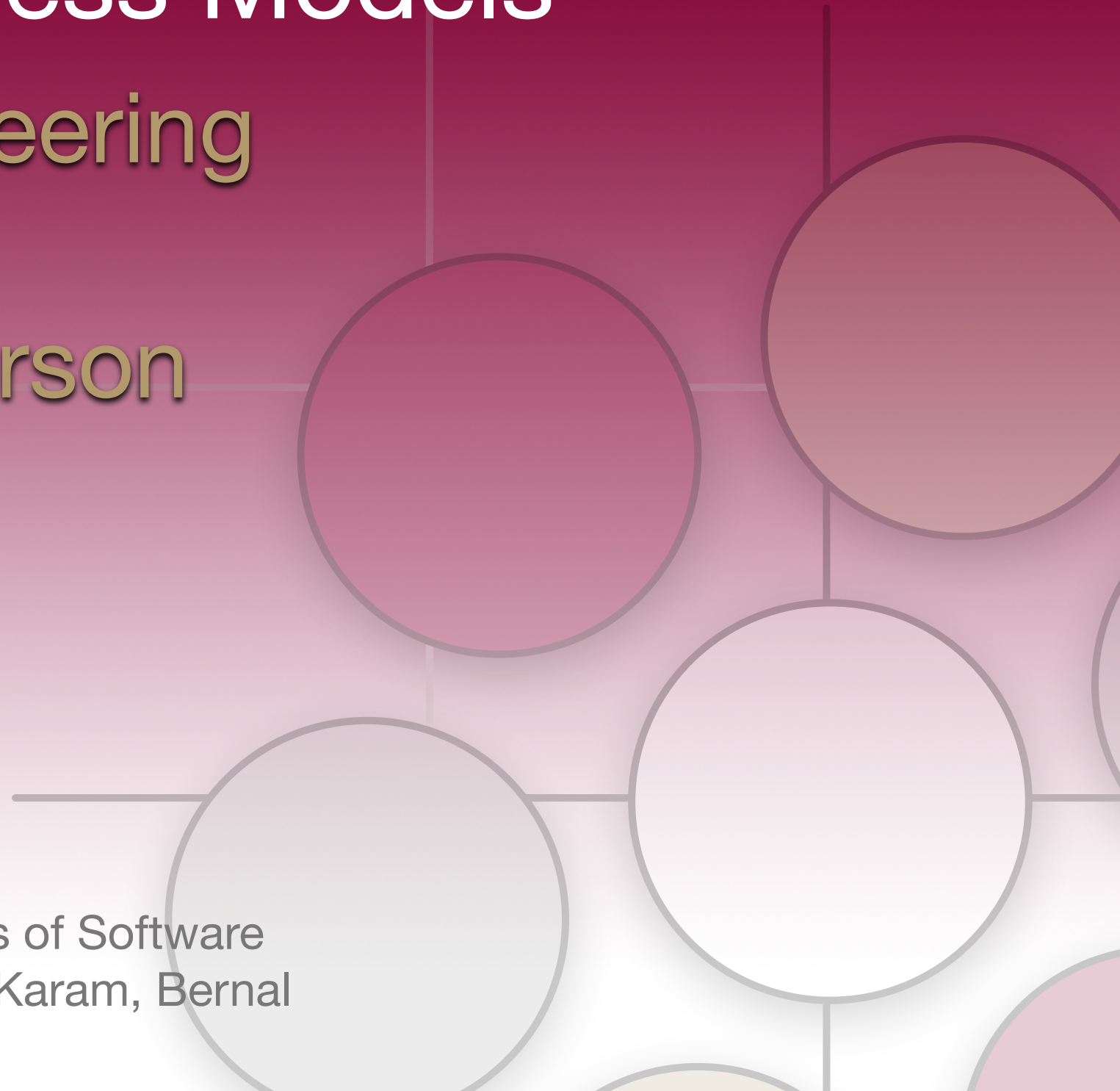
Software Process Models

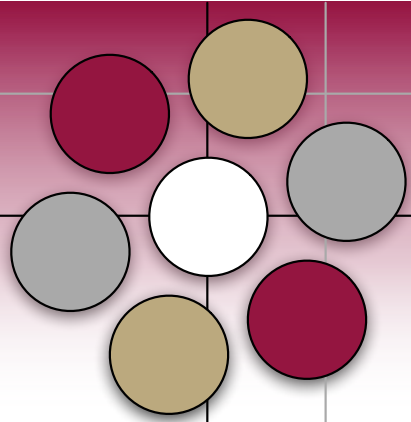
Software Engineering

CS 130

Donald J. Patterson

Content adapted from Essentials of Software Engineering 3rd edition by Tsui, Karam, Bernal Jones and Bartlett Learning

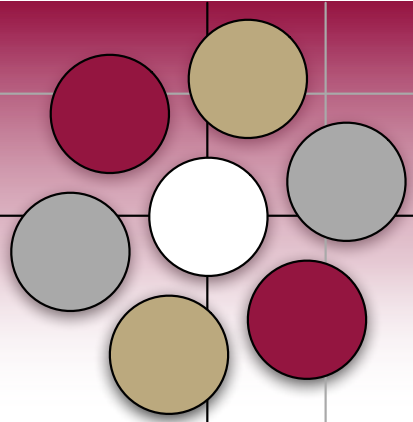




What is a **Process Model** ?

It is a **description** of

- i) what tasks need to be performed in***
- ii) what sequence under***
- iii) what conditions by***
- iv) whom to***
achieve the “desired results.”



Why Have A Process Model?

- Provide “**guidance**” for a systematic coordination and controlling of
 - a) the **tasks** and of
 - b) the **personnel** who perform the tasks

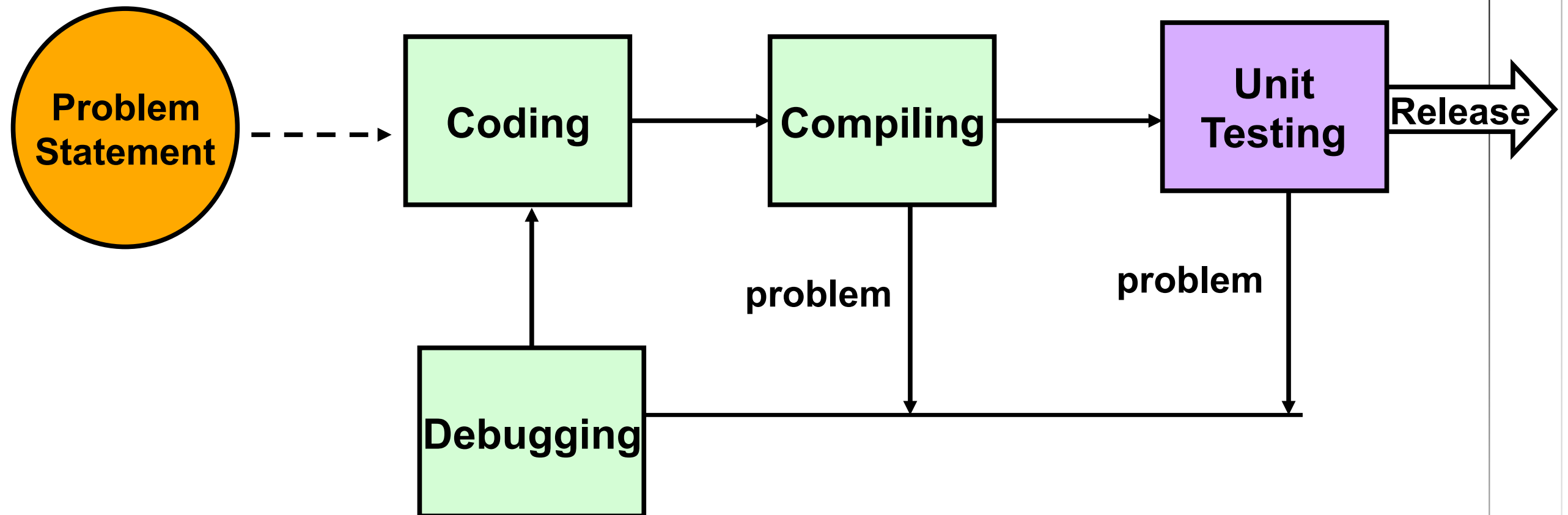
Note the key words: coordination/control, tasks, people



**Do we need a process if the project
requires just 1 person or at most
two people?**

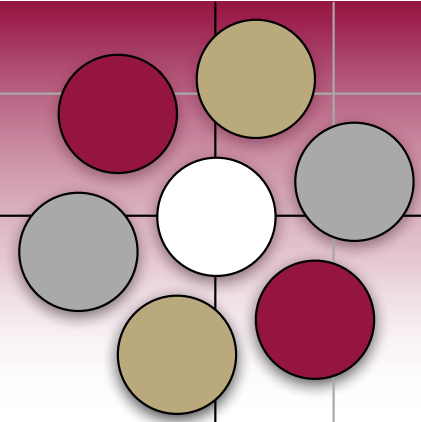
Why? -- Why not ?

A “Simple and Familiar” Process



1. Most people performs and follow this simple process, but unfortunately some skips unit testing or debugging.

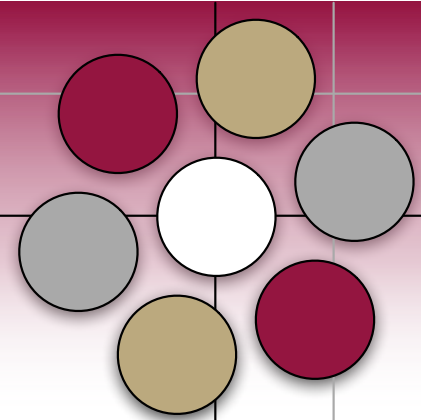
2. Also, some proceeds without thoroughly considering & understanding the “problem statement” ---- which is the requirement



Extending the “Simple” Process

- As projects got larger and more complex.
(earlier, we introduced “simplification”, “better tools”, & “process”)
 - Needed to **clarify and stabilize the requirements**
 - Needed to **test more functionalities**
 - Needed to **design more carefully**
 - Needed to **use more existing software & tools**
 - Database
 - Network
 - Code control
 - Needed **more people** to be involved

Resulting in more tasks and more people



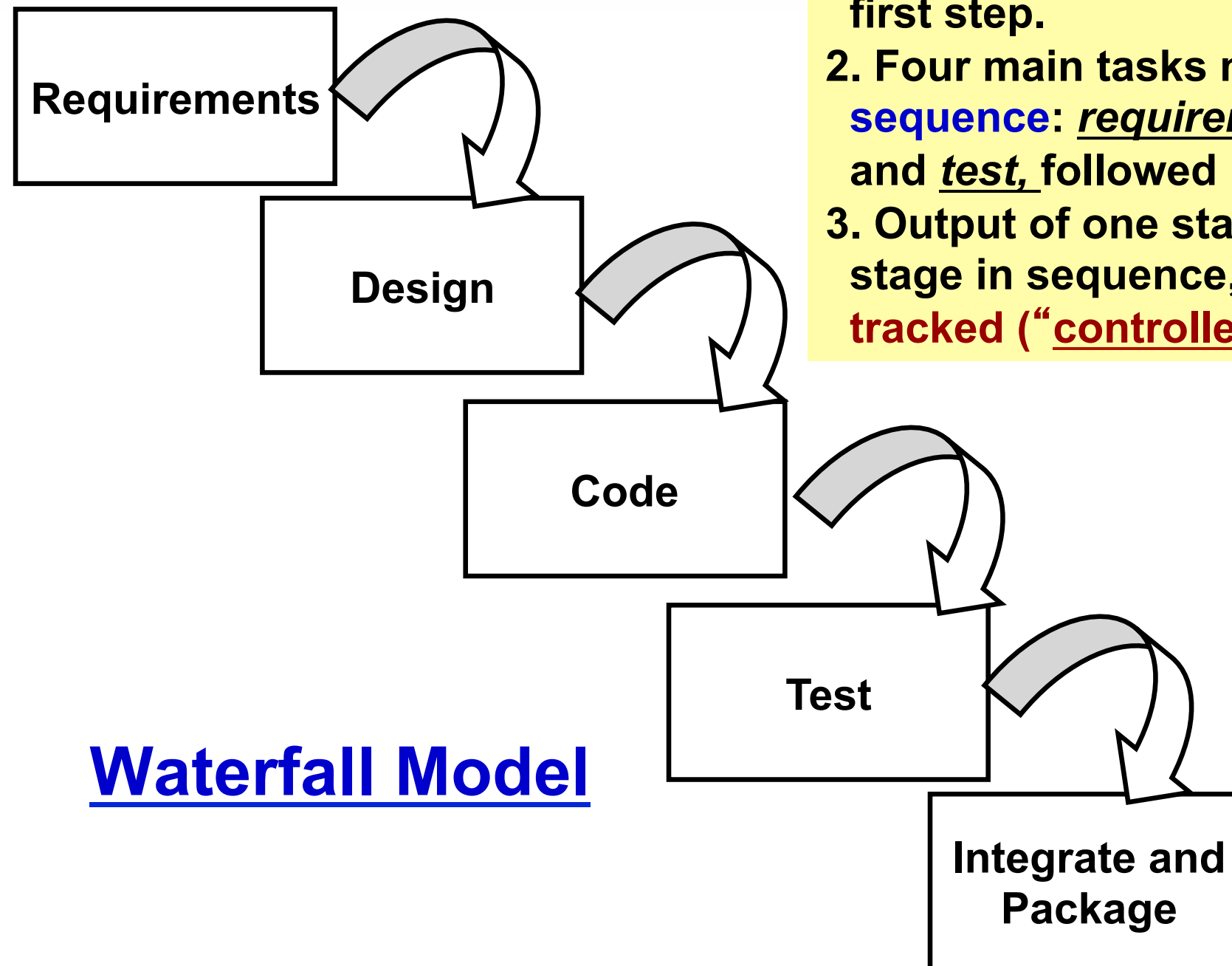
With More People and More Tasks

- We now need to **Define**:
 - the **set of tasks** that need to be performed
 - the **sequence of flow** of the tasks
 - the **input** and the **output** from these tasks
 - the **pre-condition** and **post-conditions** for each task
 - The **people & skills** needed to perform the tasks



Some “traditional” software development processes

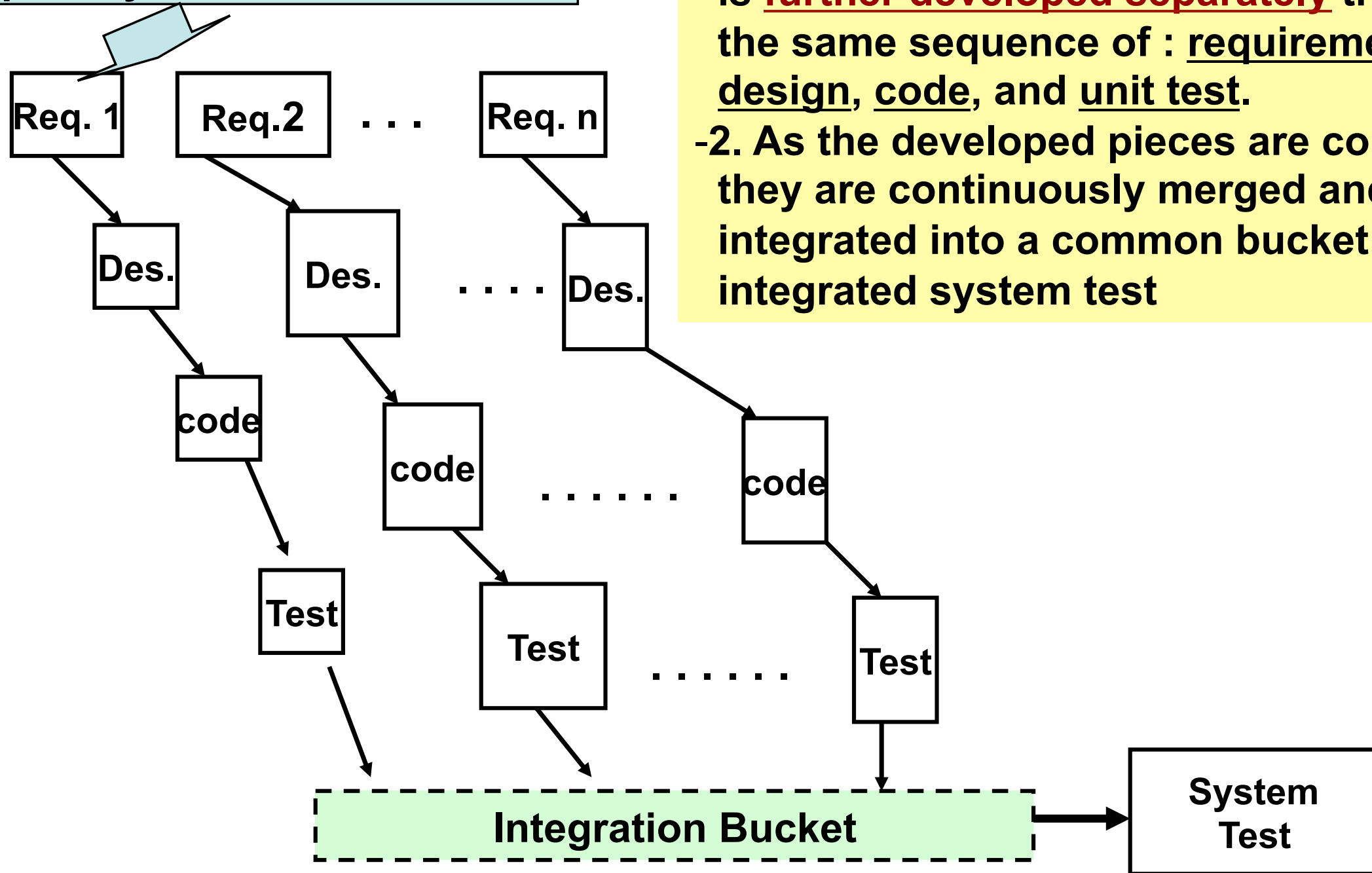
- The earlier “simple” process was employed by many for years without formally embracing other important development activities **such as requirements analysis, design, formal testing, or packaging.**
- The recognition of the need for **formal processes** was initially driven by **failures in developing large complex software --- (later shown by Chaos reports)**
 - **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 different task and managing through project phases*



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. Output of one stage feeds into the next stage in sequence, and thus **easily tracked (“controlled”)** by management

Req. Analysis and Architecture



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

Incremental Model (A)– “Continuous Integration”

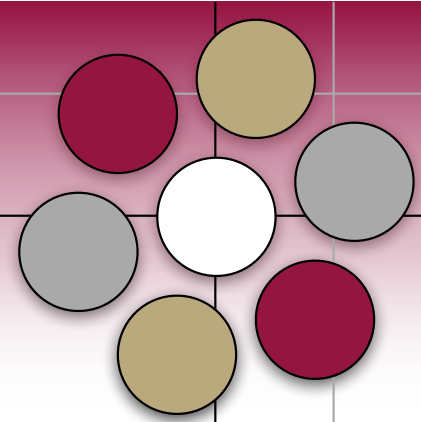


■
■
■



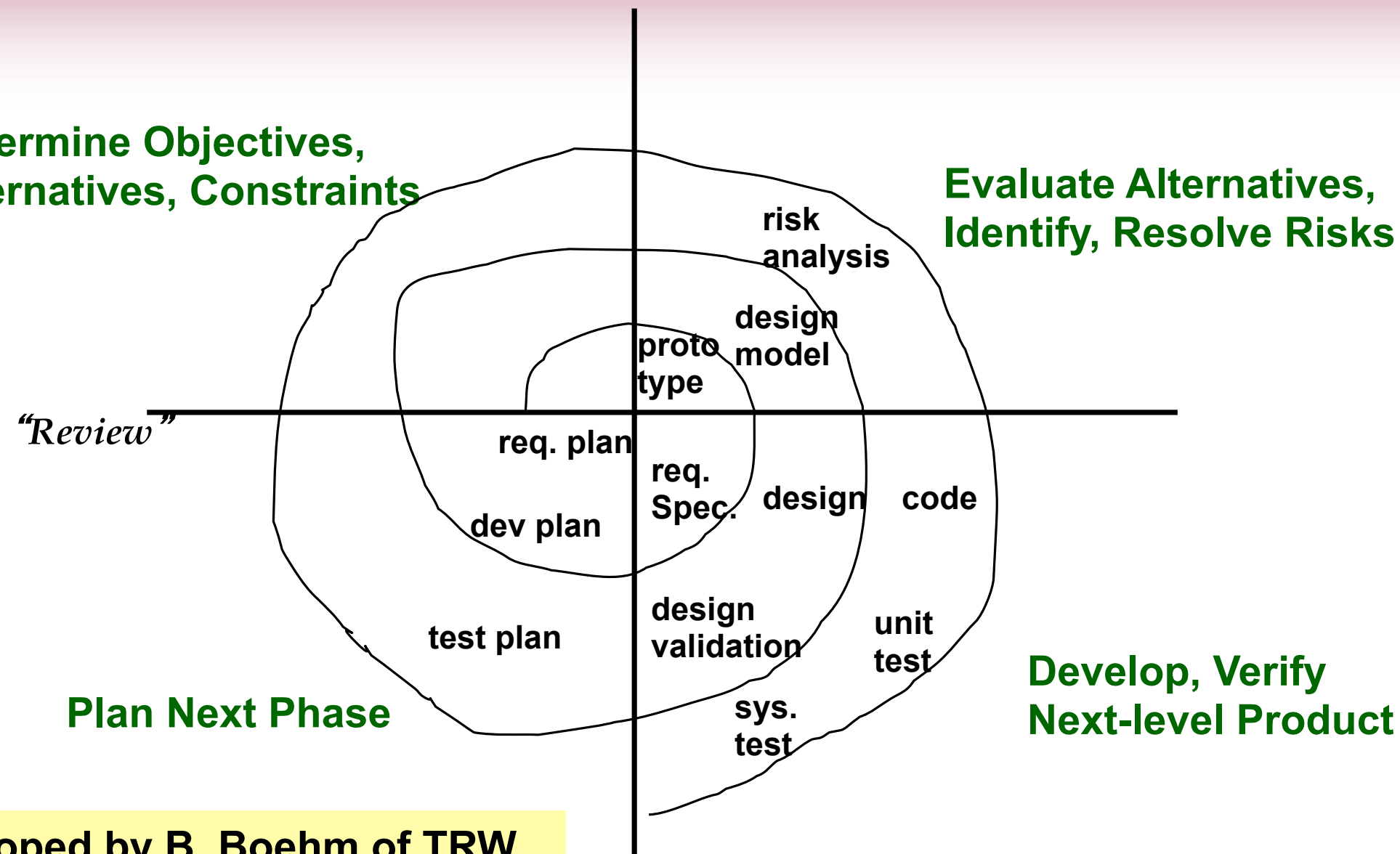
Each small set of requirements is developed, packages, and released in a **multiple release fashion**.

Incremental Model (B) - “Multiple Releases”
(seed for today’s “Agile” processes)



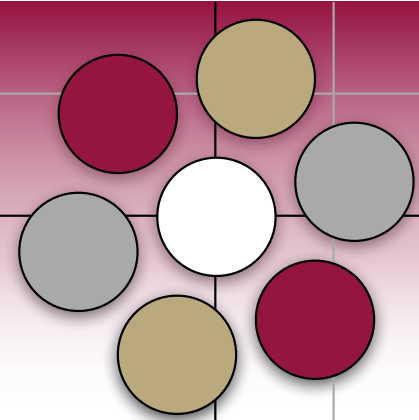
**Determine Objectives,
Alternatives, Constraints**

**Evaluate Alternatives,
Identify, Resolve Risks**



- developed by B. Boehm of TRW
- Software development activities are cycled through 4 phases
- A “**Risk Averse**” process first proposed by Barry Boehm

Spiral Model



Phases of Project

Activities

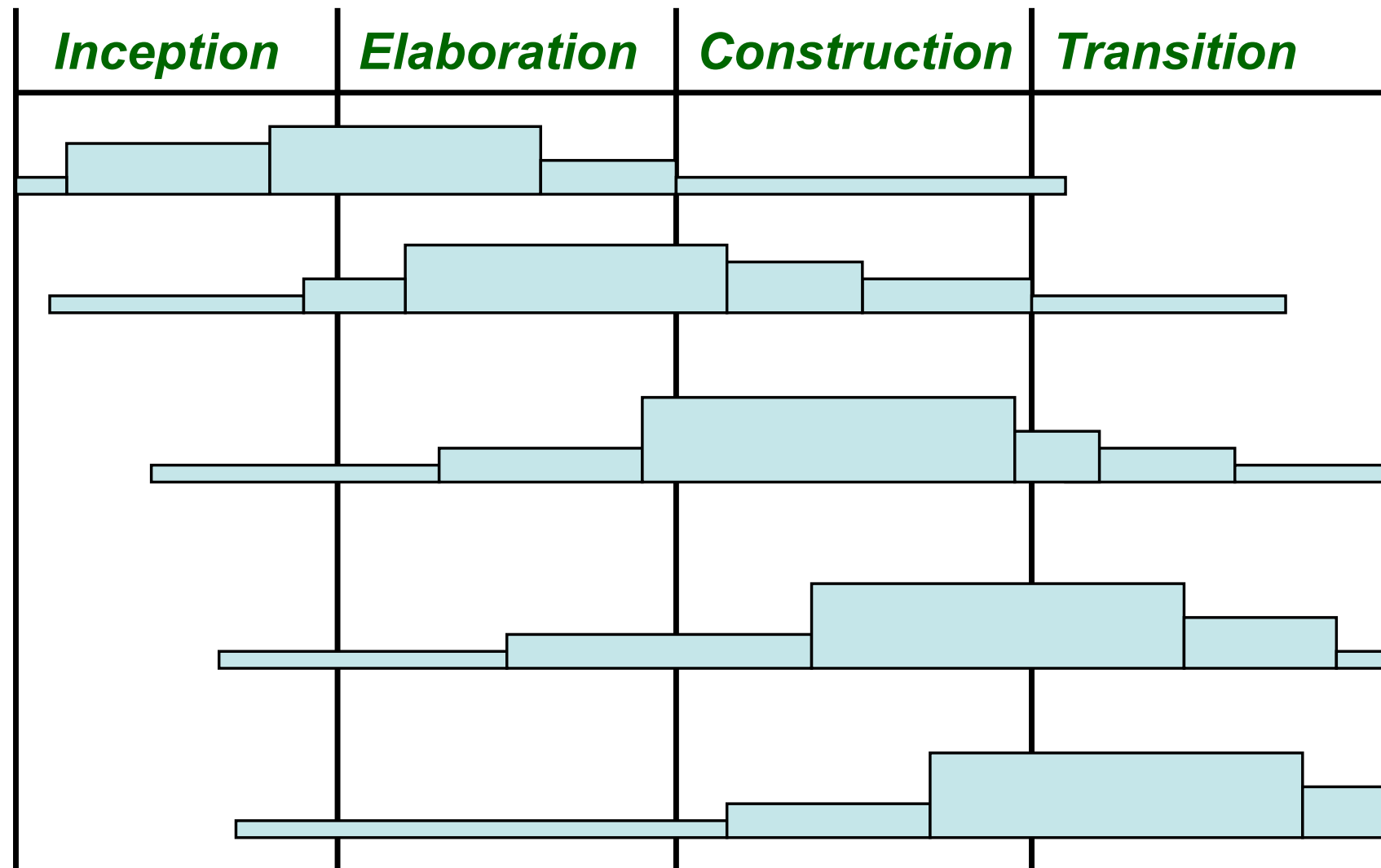
Requirements

Design

Implement

Test

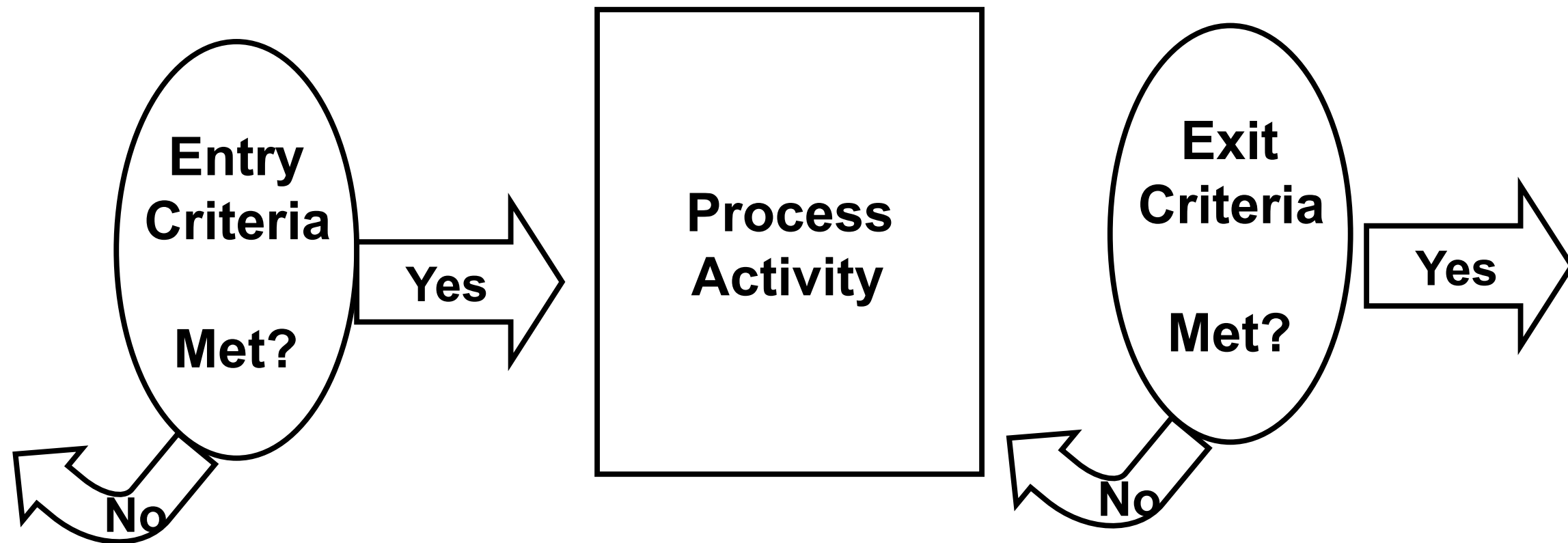
Integrate



Rational Unified Process (RUP)

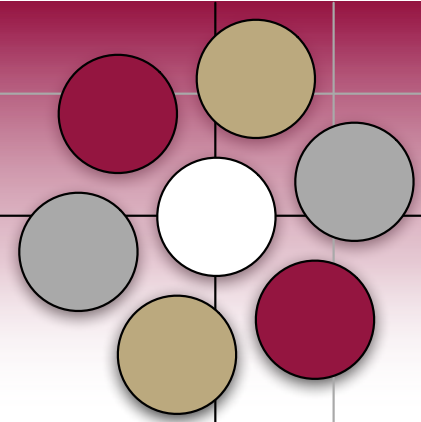
Every software development activity is “addressed” in the 4 phases of inception, elaboration, construction, and transition

Entry and Exit Criteria



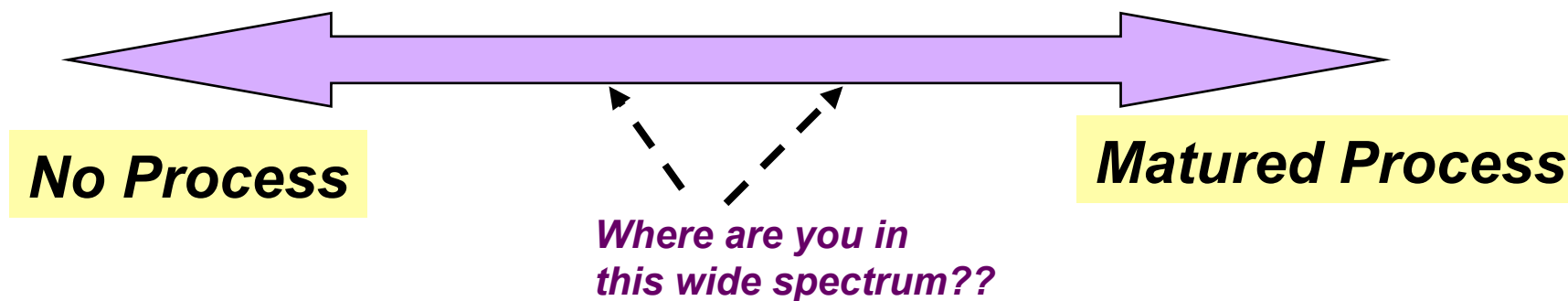
In order for process models to be more than just a “guideline,” it must include a list of conditions or requirements that define the:

- **entry criteria** prior to performing an activity in a process.
- **exit criteria** before an activity in the process is deemed completed.

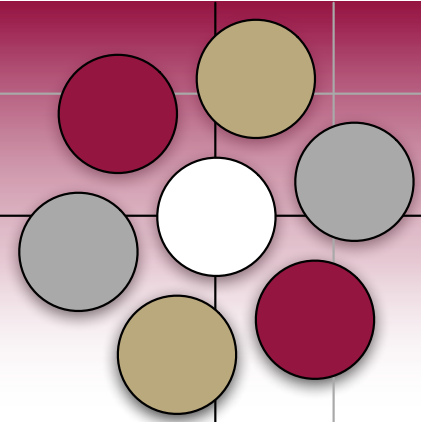


Assessment of Software Organizations

- Software Development and Software Support may be done with very little process or with very sophisticated, well defined, well organized and well executed processes.
- How mature is your software engineering organization and do you need to improve?
- ISO (ISO 9000 series) and SEI (Software Engineering Institute at Carnegie Mellon) are two leading organizations that help in the process assessment

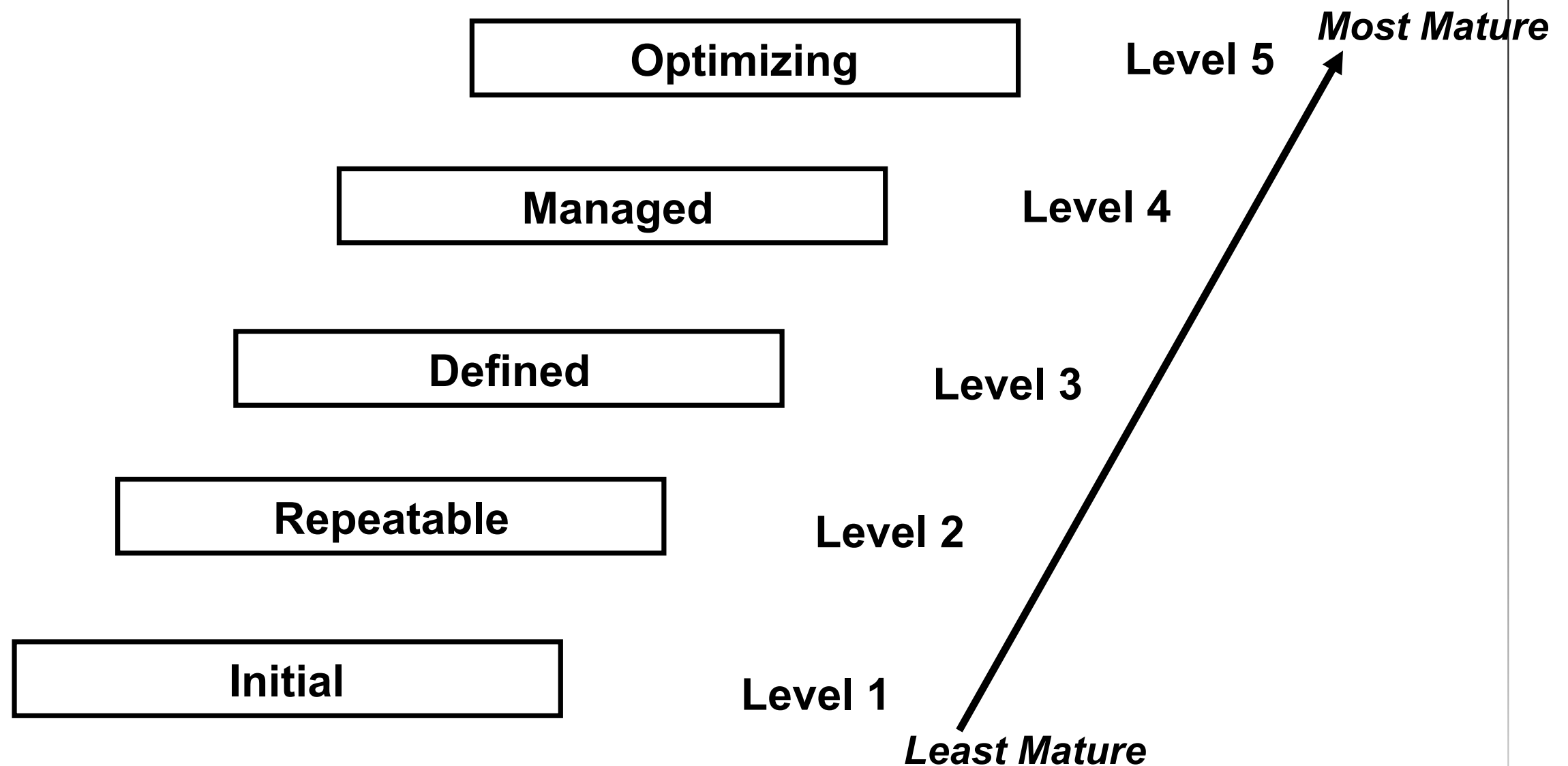


SEI's Original CMM – Early 1990s



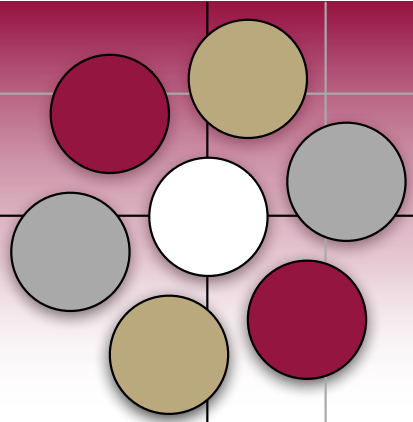
- Software Engineering Institute (SEI) proposed a Capability Maturity Model (CMM) to help software organizations assess their maturity and provide guidance in software development.
 - Initial: there is **no process** and any success is by luck or with a special person.
 - Repeatable: has mastered **6 processes** and can repeat its success with these 6 processes: 1) requirements mgmt, 2) project tracking, 3) quality assurance, 4) project planning, 5) subcontract mgmt, and 6) configuration management
 - Defined: has mastered **7 more processes** and is competent at software construction: 1) organization process, 2) training program, 3) product engineering, 4) peer review, 5) organization process definition, 6) integrated soft. mgmt, and 7) inter-group coordination
 - Managed: has introduced **2 more processes** that deal with quantitative measurement and quality: 1) quantitative process management and 2) quality mgmt
 - Optimizing: reaching this highest level requires the mastering of continuous improvement with **3 more processes**: 1) defect prevention, 2) technology change management, 3) process change management

SEI's 5 Levels of Original "Capability Maturity Model" (CMM)



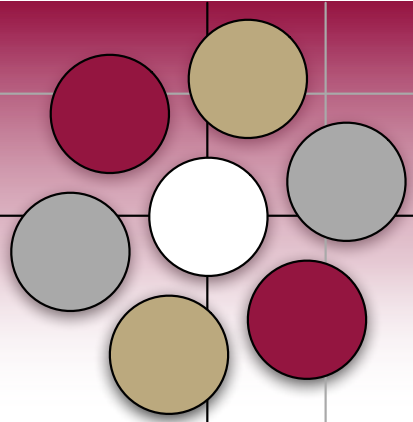
Total of 18 processes need to be mastered to achieve "optimized" level

See page 92 of your text for the 18 "key" processes



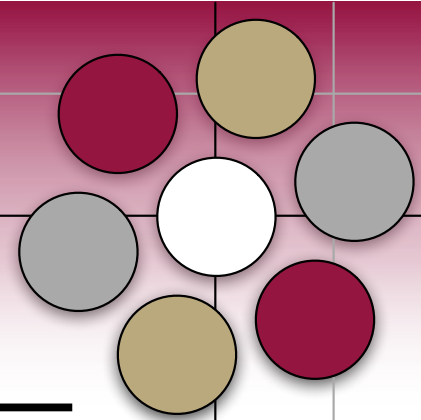
SEI' s CMMI

- In 2001, CMM was upgraded to **CMMI (CMM Integrated)**. Started with multiple, major aspects to CMMI:
 - Systems engineering
 - **Software engineering**
 - Integrated product and process development
 - Supplier sourcing



2 Representations of CMMI

- The **software engineering** portion of **CMMI** has two representations:
 - **Staged** : similar to the CMM assessment of organization
 - **Continuous** : better for assessing maturity of each process

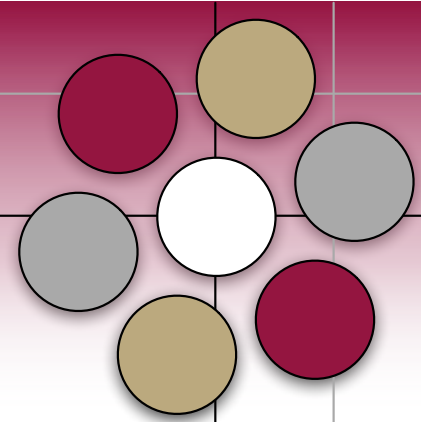


Level 5	Optimizing	Optimizing
Level 4	Quantitatively Managed	Quantitatively Managed
Level 3	Defined	Defined
Level 2	Managed	Managed
Level 1	Performed	Initial
Level 0	Incomplete	- - - - -

Continuous
(Capability Levels)

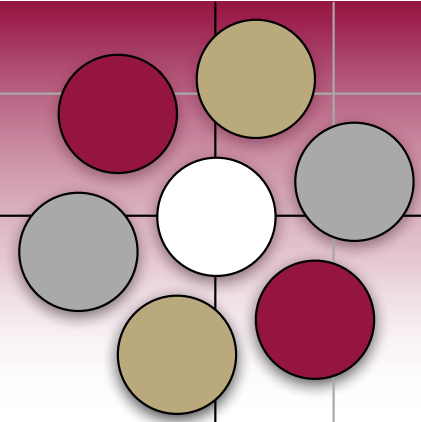
Staged
(Maturity Levels)

Levels for Continuous versus Staged models in CMM I



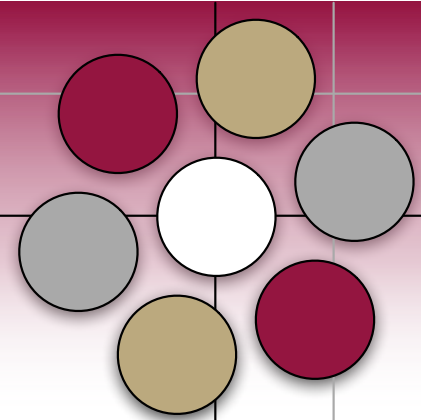
25 Processes of CMMI

- There are **25 processes** covering **4 major categories** :
 - **Process Management** (has 5 processes):
 - Organization process focus
 - Organizational process definition
 - Organizational training
 - Organizational process performance
 - Organizational innovation and deployment
 - **Project Management** (has 8 processes):
 - Project planning
 - Project monitoring and control
 - Supplier agreement management
 - Integrated project management
 - Risk management
 - Integrated teaming
 - Integrated supplier management
 - Quantitative project management



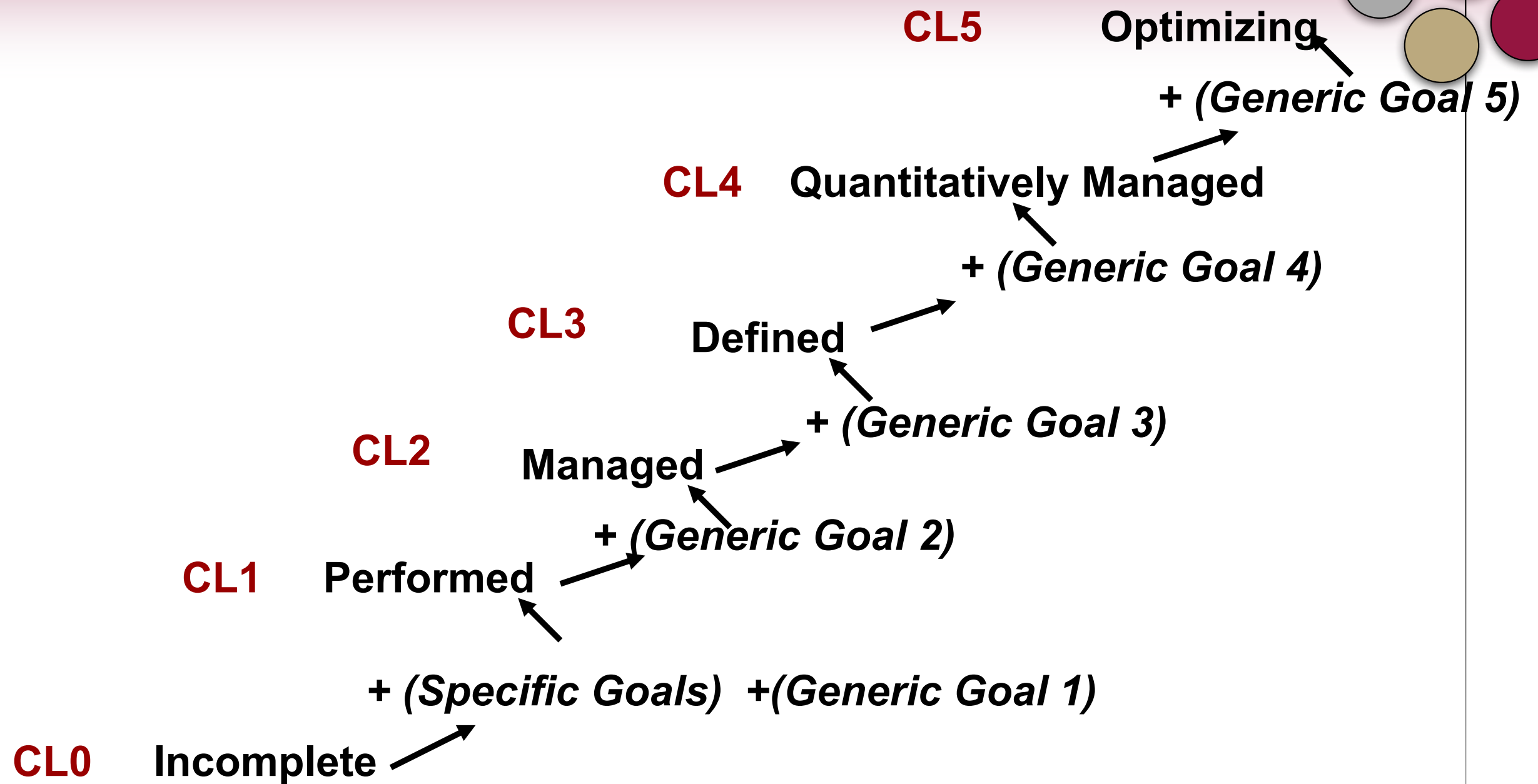
25 Processes of CMMI (cont,)

- **Engineering** (has 6 processes)
 - Requirements development
 - Requirements management
 - Technical solution
 - Product integration
 - Verification
 - Validation
- **Support** (has 6 processes)
 - Configuration management
 - Process and product quality assurance
 - Measurement and analysis
 - Organizational environment for integration
 - Decision analysis and resolution
 - Causal analysis and resolution

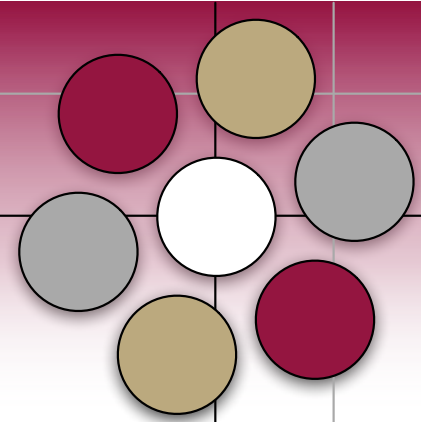


Continuous versus Staged Models

- In **Continuous Representation**, each process starts at capability level 0 and moves up the capability levels based on achieving “generic goals” and “specific sub-goals.”
 - Allows the organization to **choose and pick the process to focus on** based on the needs of the organization
 - Allows **comparison of process area by process area** between organizations
 - Allows easier migration from other standards
- In **Staged Representation**, the organization starts at maturity level 1 and moves up the levels based on mastering sets of processes.
 - Allows **easy migration from the earlier CMM model**
 - Provides a guidance of sequence of maturity by process areas
 - Allows **easier comparison of organizations by maturity levels**

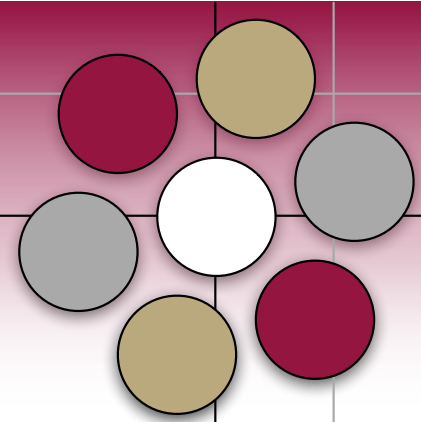


Achieving the “*Capability Levels*” by each Process Area in the **Continuous Representation Model**



5 Generic Goals

- **Goal 1 – Achieve all the specific goals of the specific process**
- **Goal 2 – *Institutionalize* the managing of consistency of that process across organization**
- **Goal 3 – *Institutionalize* the defining of that process across the organization**
- **Goal 4 – *Institutionalize* quantitatively managing that process across the organization**
- **Goal 5 - *Institutionalize* continuous optimizing/improving that process across the organization**

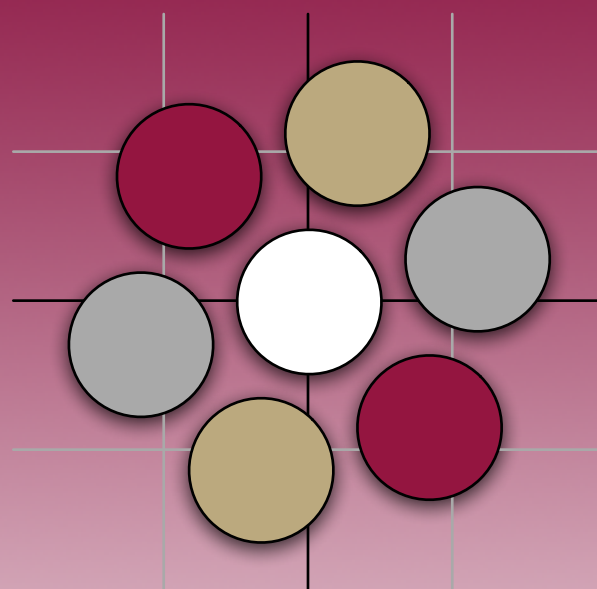


Achieving “*Maturity Level*” (ML) in the Staged Representation model

- **ML1** (0 process) : no process
- **ML2** (7 processes): 1)Requirements Mgmt, 2)Project planning, 3)Project monitoring, 4)Supplier agreement mgmt, 5)Measurement and analysis, 6)Process and product quality assurance, 7)Configuration mgmt
- **ML3** (14 processes): 1)Requirements development, 2)Technical solution, 3)Product integration, 4)Verification, 5)Validation, 6)Organizational process focus, 7)Organizational process definition, 8)Organizational training, 9)Integrated project management, 10)Risk management, 11)Integrated teaming, 12)Integrated supplier mgmt, 13)Decision analysis and resolution, 14)Organizational environment for integration
- **ML4** (2 processes): 1)Organizational process performance, 2)Quantitative project management
- **ML5** (2 processes): 1)Organizational innovation and deployment, 2)Causal analysis and resolution

Process Definition & Communication (extra)

- Recap - 2 Main Components of Process Definition:
 - Major activities
 - Sequencing of activities
- *Most of the organizations need to modify an existing process to better fit their needs ----- thus they must define in more detail and communicate the modified process definitions (a major endeavor)*
- Expanding process definition to more “refined” level:
 - Detailed description of the activities
 - Control needed for *entrance* and *exit* of each activity and the ordering of the activities
 - Artifacts that result from the activities
 - Human resources required to perform the activities
 - Tools that may be needed to aid the performance of the activities



WESTMONT COMPUTER SCIENCE