Software Project Management
Sixth Edition

Chapter 13.2

Software process quality

Product and Process Quality

- A good process is usually required to produce a good product.
- For manufactured goods, process is the principal quality determinant.
- For design-based activity, other factors are also involved:
  - For example, the capabilities of the designers.

BS EN ISO 9001:2000 and quality management systems

- ISO 9001 is one of a family of standards that specify the characteristics of a good quality management system (QMS)
- Can be applied to the creation of any type of product or service, not just IT and software
- Does NOT set universal product/service standards
- DOES specify the way in which standards are established and monitored

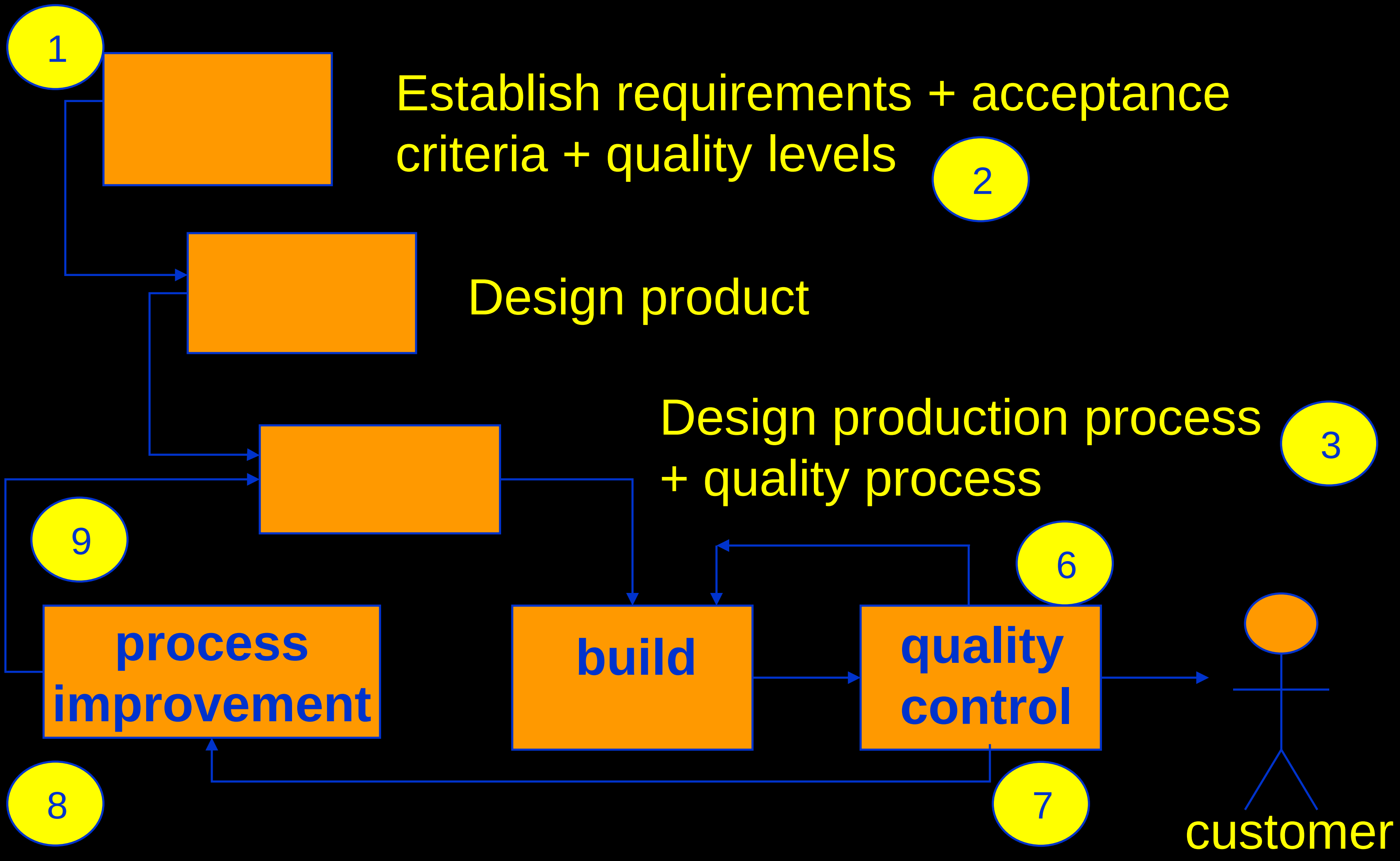
ISO 9001:2000 principles

1. Understanding the requirements of the customer
2. Leadership to provide unity of purpose and direction to achieve quality
3. Involvement of staff at all levels
4. Focus on the individual which create intermediate and deliverable products and services

ISO 9001:2000 principles

5. Focus on interrelation of processes that deliver products and services
6. Continuous process improvement
7. Decision-making based on factual evidence
8. Mutually beneficial relationships with suppliers

ISO 9001:2000 cycle


1
Establish requirements + acceptance criteria + quality levels
2
Design product
Design production process + quality process
3
9
6
process improvement
build
quality control
customer
8
7

SPM (6e) Software process quality© The McGraw-Hill Companies, 2017

ISO 9001 Requirements

- Management responsibility
- Quality system
- Contract review
- Design Control
- Document and data control
- Purchasing
- Control of customer supplied product
- Product identification and traceability
- Process control
- Inspection and testing
- Control of inspection, measuring and test equipment

The need to improve

- can everything be improved at one?

  no, must tackle the most important things first

- ‘poor companies are poor at changing’

  some later improvements build on earlier ones

- but there are problems
  - *improvement takes up time and money*
  - *‘improvement’ may simply be more bureaucracy!*

Capability maturity model (CMM)

- Created by Software Engineering Institute, Carnegie Mellon University
- CMM developed by SEI for US government to help procurement
- The rationale was:
  - Include likely contractor performance as a factor in contract awards.
- Watts S. Humphrey 'Managing the software process' Addison Wesley
- Assessment is by questionnaire and interview

Capability maturity model 2

- Different versions have been developed for different environments e.g. software engineering
- New version CMMI tries to set up a generic model which can be used for different environments

SEI Capability Maturity Model

- Can be used in two ways:
  - Capability evaluation
  - Software process assessment.

Capability Evaluation

- Provides a way to assess the software process capability of an organization:
  - Helps in selecting a contractor
  - Indicates the likely contractor performance.

Software Process Assessment

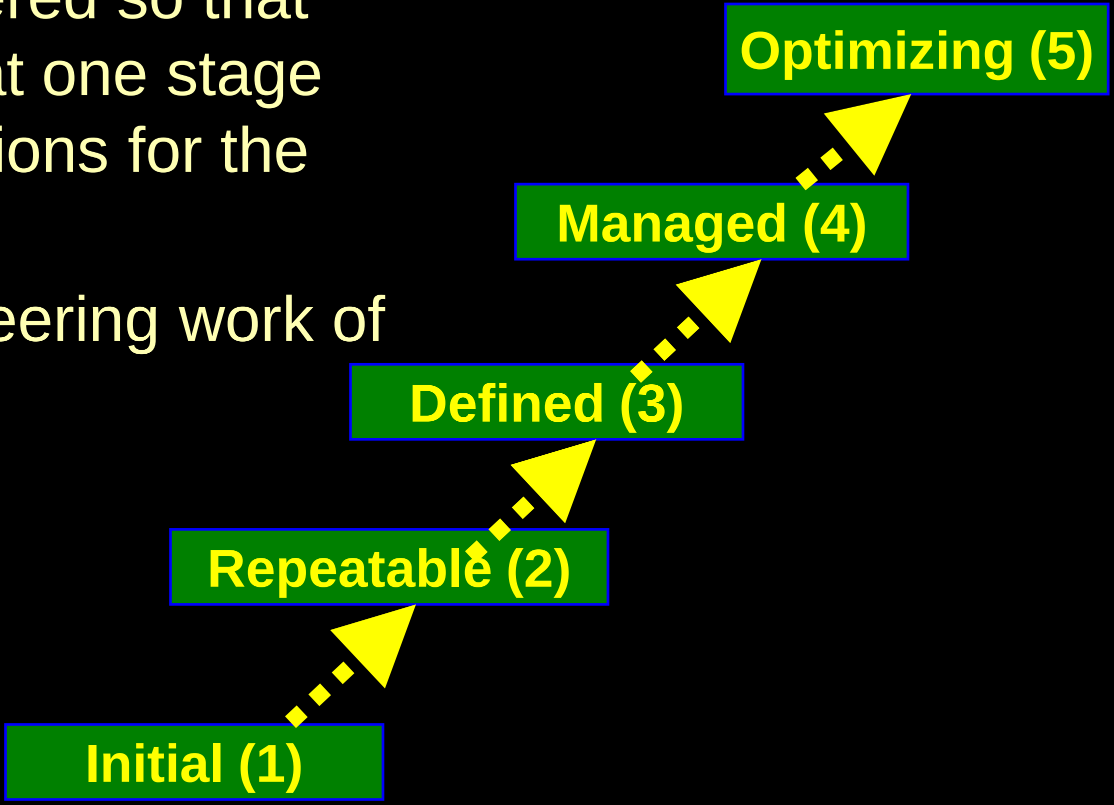
- Used by an organization to assess its current process:
  - Suggests ways to improve the process capability.
- This type of assessment is for purely internal use.

What is CMM?

- Describes an evolutionary improvement path for software organizations from an ad hoc immature process :
  - To a mature, disciplined one.
- Provides guidance on:
  - How to control the process
  - How to evolve the process

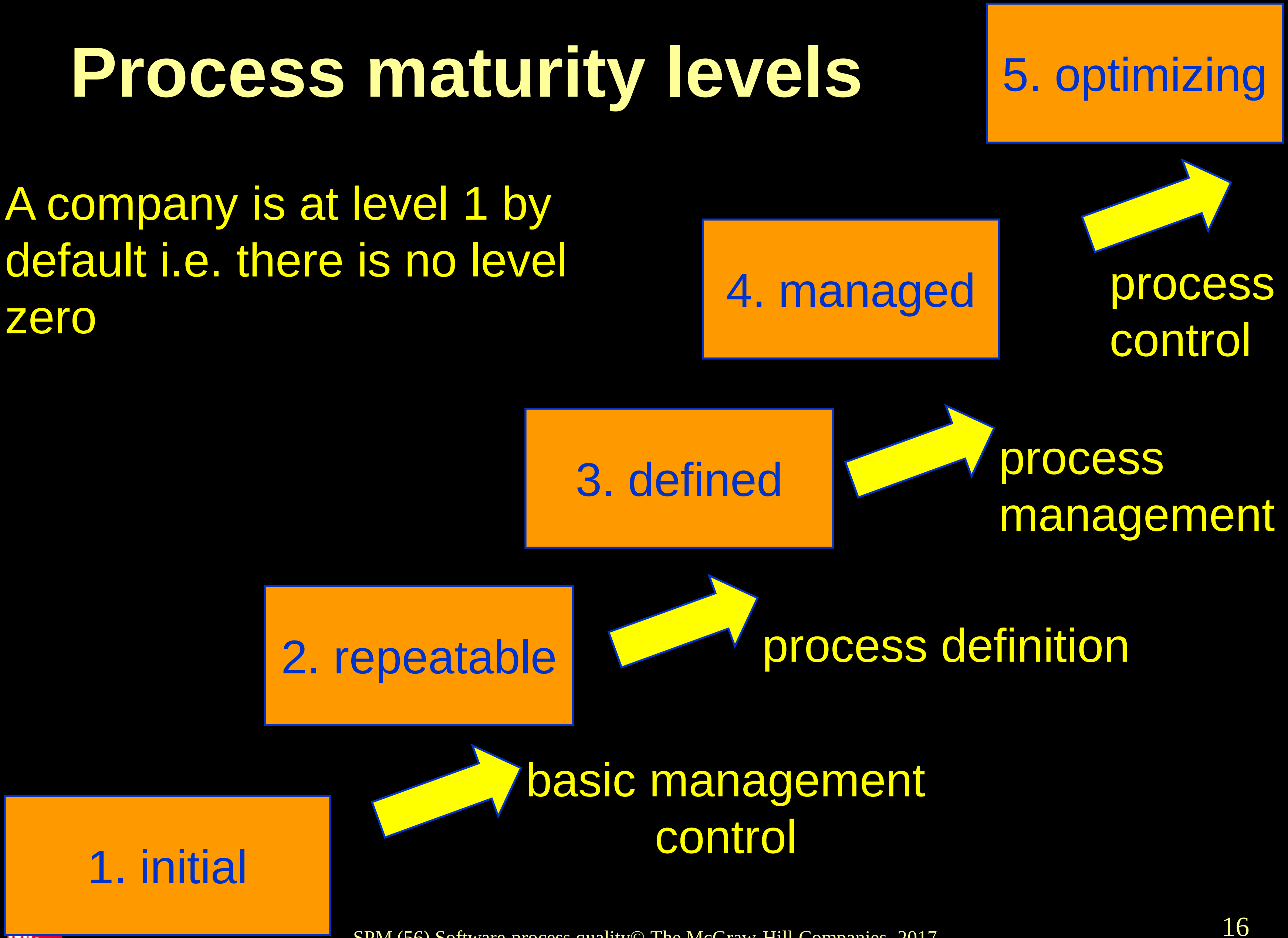
CMM Maturity Levels

- Five maturity levels:
  - Stages are ordered so that improvements at one stage provide foundations for the next.
- Based on the pioneering work of Philip Crosby.



Process maturity levels

A company is at level 1 by default i.e. there is no level zero



5. optimizing

process control

4. managed

process management

3. defined

process definition

2. repeatable

basic management control

1. initial

CMM Level 1 (Initial)

- Organization operates Without any formalized process or project plans
- An organization at this level is characterized by Ad hoc and chaotic activities.
- Software development processes are not defined,
- Different developers follow their own process
- The success of projects depend on individual efforts and heroics.

Level 2 (Repeatable)

- Basic project management practices are followed
  - Size and cost estimation techniques:
  - Function point analysis, COCOMO, etc.
  - Tracking cost, schedule, and functionality.
- Development process is ad hoc:
  - Not formally defined
  - Also not documented.

Level 3 (Defined)

- All management and development activities:
  - Defined and documented.
  - Common organization-wide understanding of activities, roles, and responsibilities.
- The process though defined:
  - Process and product qualities are not measured.

Level 4 (Managed)

- Quantitative quality goals for products are set.
- Software process and product quality are measured:
  - The measured values are used to control the product quality.
- Results of measurement used to evaluate project performance:
  - Rather than improve process.
- Detailed measures of the software process and product quality are collected.
- Both the software process and products are quantitatively understood and controlled.

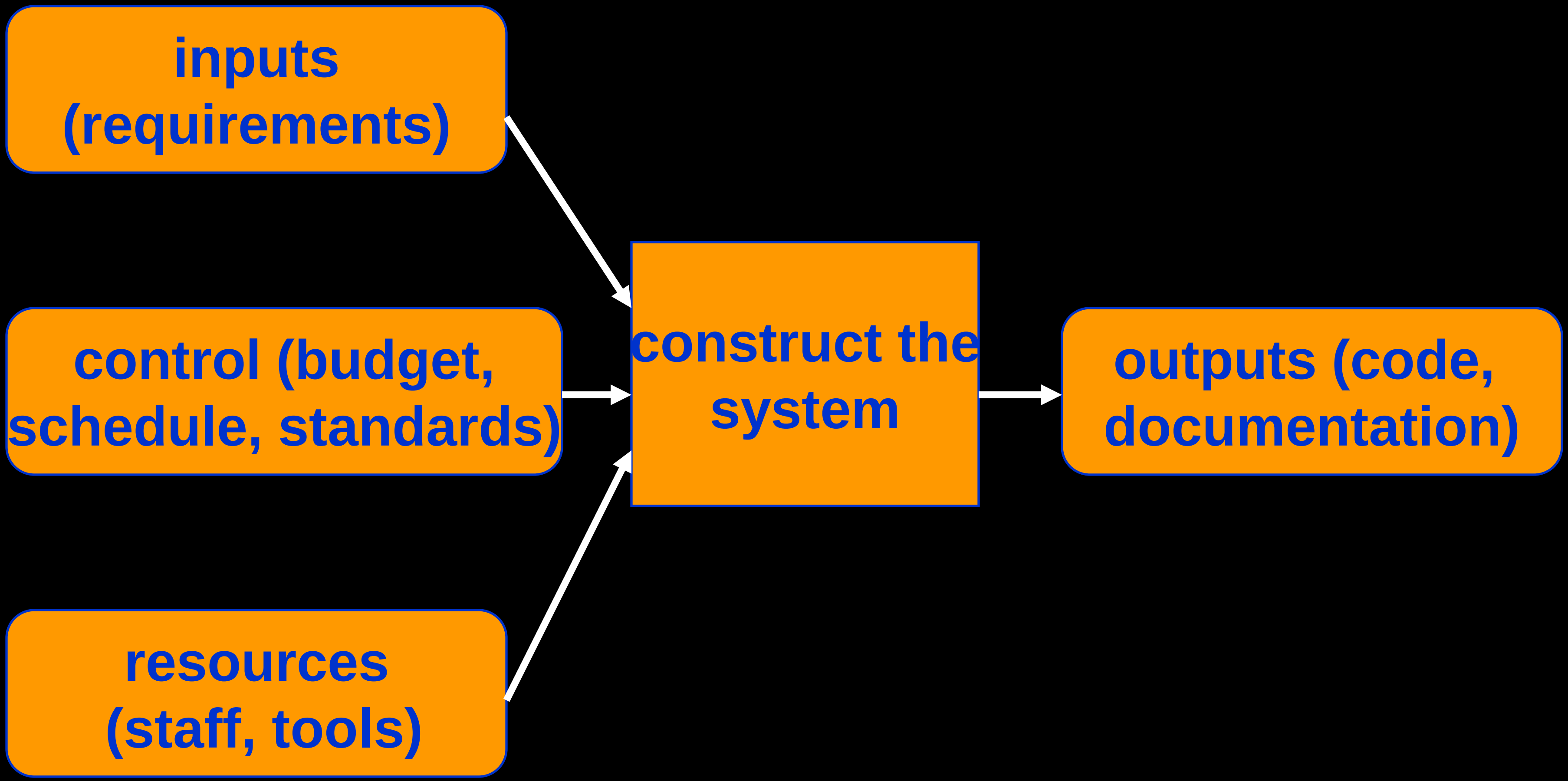
Level 5 (Optimizing)

- Statistics collected from process and product measurements are analyzed:
  - Continuous process improvement based on the measurements.
- Known types of defects are prevented from recurring by tuning the process
- Lessons learned from specific projects incorporated into the process

Key process areas

- The KPAs of a level indicate the areas that an organization at the lower maturity level needs to focus to reach this level.
- KPAs provide a way for an organization to gradually improve its quality of over several stages.
- KPAs for each stage has been carefully designed such that one stage enhances the capability already built up.
  - Trying to focus on some higher level KPAs without achieving the lower level KPAs would be counterproductive.

A repeatable model

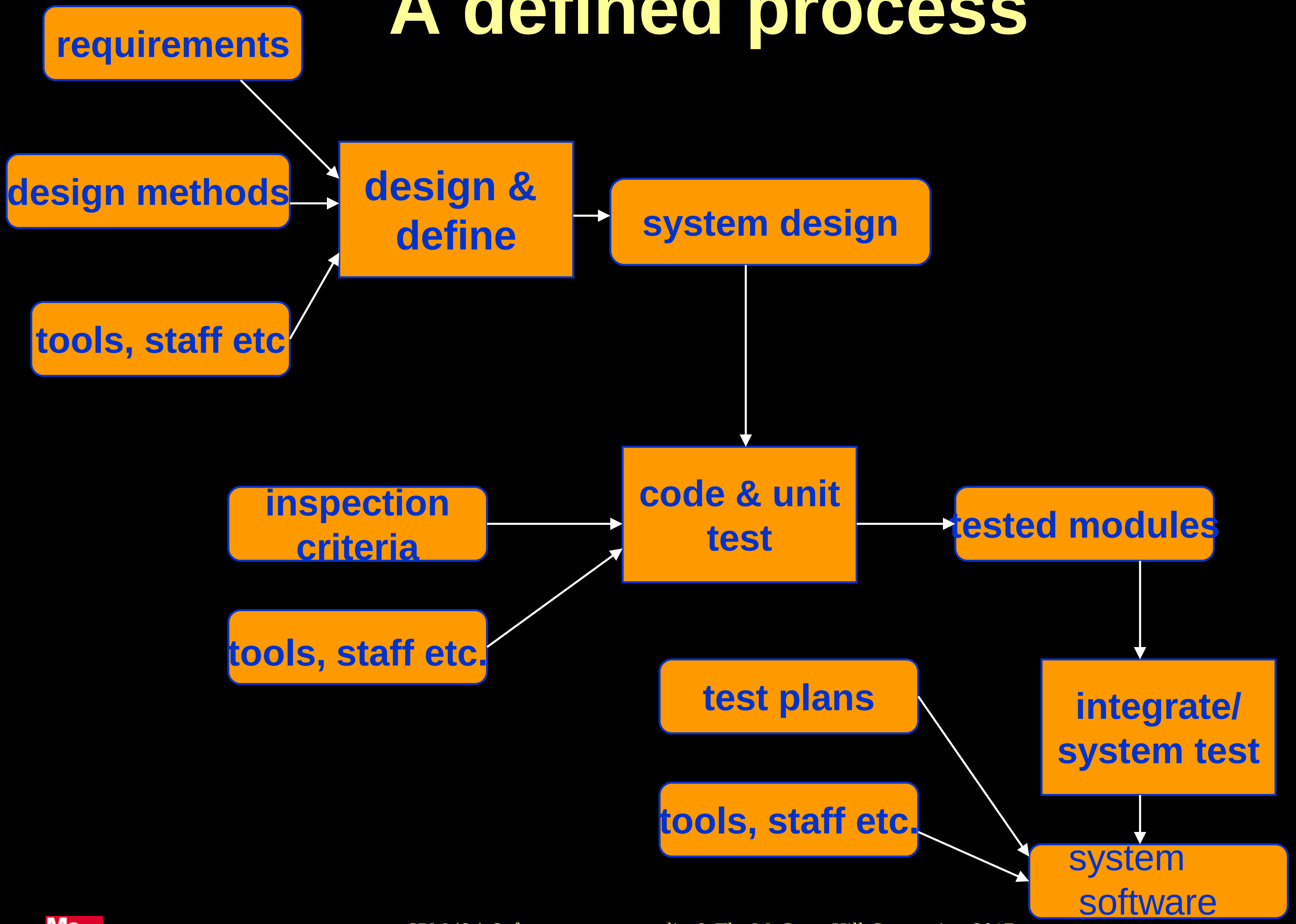

inputs (requirements)
control (budget, schedule, standards)
resources (staff, tools)
construct the system
outputs (code, documentation)

Repeatable model KPAs

To move to this level concentrate on:

- Configuration management
- Quality assurance
- Sub-contract management
- Project planning
- Project tracking and oversight
- Measurement and analysis

A defined process

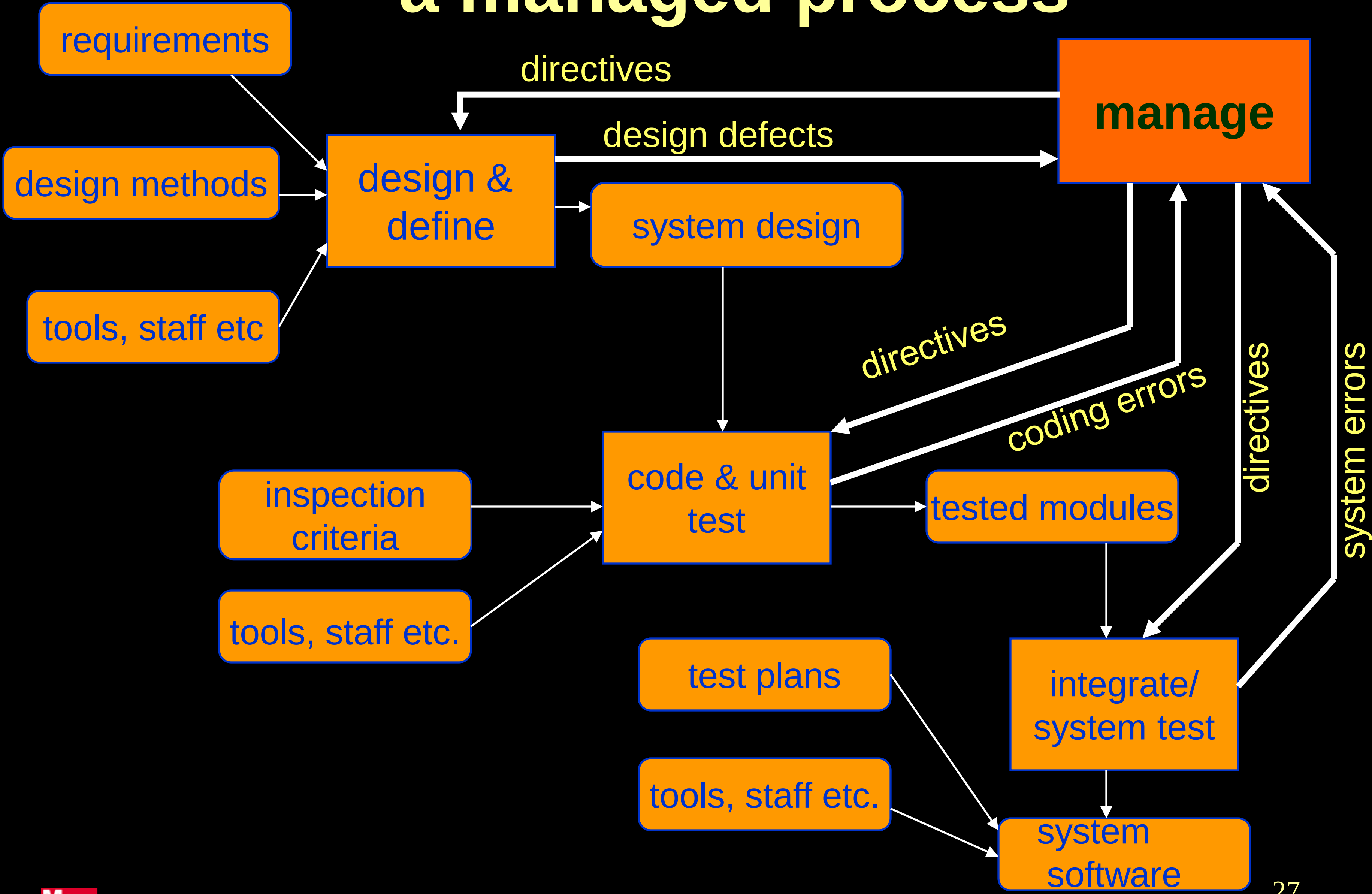


Repeatable to defined KPAs

Concentrate on

- Requirements development and technical solution
- Verification and validation
- Product integration
- Risk management
- Organizational training
- Organizational process focus (function)
- Decision analysis and resolution
- Process definition
- Integrated project management

a managed process

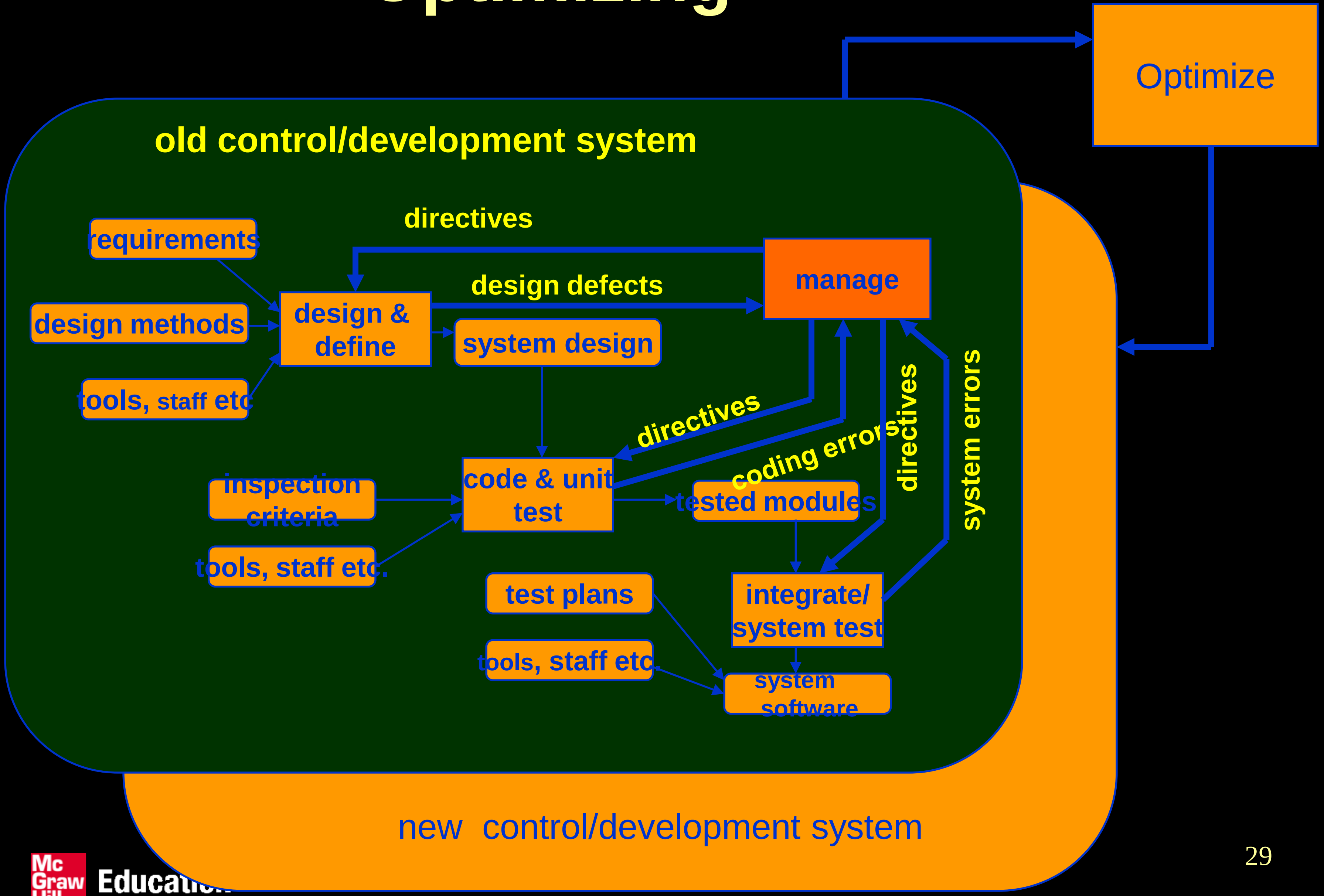


Defined to managed KPAs

Concentrate on:

- Organizational process performance
- Quantitative project management

Optimizing



Optimize

old control/development system

directives

requirements

design methods

tools, staff etc

design & define

design defects

manage

system design

directives

coding errors

directives

system errors

inspection criteria

tools, staff etc.

code & unit test

tested modules

test plans

tools, staff etc.

integrate/ system test

system software

new control/development system

Managing to optimizing: KPAs

Concentrate on:

- Causal analysis and resolution
- Organizational innovation and deployment

CMMI (Capability Maturity Model Integration)

- CMMI is the successor of the Capability Maturity Model (CMM).
- After CMMI was first released in 1990:
  - It became popular in many other domains
  - Human Resource Management (HRM).: people management (PCMM)
  - software acquisition (SA-CMM)
  - systems engineering (SE-CMM)

Some questions about CMMI

- Suitable only for large organizations?
  - e.g. need for special quality assurance and process improvement groups
- Defining processes may not be easy with new technology
  - how can we plan when we’ve not used the development method before?
- Higher CMM levels easier with maintenance environments?
- Can you jump levels?

ISO/IEC 15504 IT process assessment

- To provide guidance on the assessment of software development processes
- **Process Reference Model**: Needs a defined set of processes that represent good practice to be the benchmark
- **ISO 12207** is the default reference model
- Could use others in specific environments

ISO 15504 performance attributes

CMMI level	ISO 15504
	0. incomplete
initial	1.1.process performance – achieves defined outcome
repeatable	2.1 process management – it is planned and monitored
	2.2 work product management – control of work products

ISO 15504 performance attributes - contd

CMMI	ISO 15504
Defined	3.1. Process definition
	3.2. Process deployment
Managed	4.1. Process measurement
	4.2. Process control
Optimizing	5.1. Process innovation
	5.2. Process optimization

Process Reference Model

- A defined standard approach to development
- Reflects recognized good practice
- A benchmark against which the processes to be assessed can be judged
- ISO 12207 is the default model

IS0 15504 Process Assessment

For each process in the relevant Process Reference Model

- For each set of attribute level criteria
  - Assess whether:
    - N: not achieved 0-15%
    - P: partially achieved >15%-50%
    - L: largely achieved >50%-85%
    - F: fully achieved >85%

ISO 15504 performance indicators

This is just an example of how indicators for each level *might* be identified

1.Performance

Descriptions of maximum and minimum expected input values exist

2.1 Performance management

A plan of how expected input variable ranges are to be obtained exists which is up to date

ISO 15504 performance indicators 2

2.2 Work product management

There are minutes of a meeting where the input requirements document was reviewed and corrections were mandated

3.1 Process definition

A written procedure for input requirements gathering exists

3.2 Process deployment

A control document exists that is signed as each part of the procedure is completed

ISO 15504 performance indicators 2

4.1. Process measurement

Collected measurement data can be collected e.g. number of changes resulting from review

4.2. Process control

Memos relating to management actions taken in the light of the above

ISO 15504 performance indicators 3

5.1 Process innovation

Existence of some kind of ‘lessons learnt’ report at the end of project

5.2. Process optimization

Existence of documents assessing the feasibility of suggested process improvements and which show consultation with relevant stakeholders

Techniques to improve quality -Inspections

- When a piece of work is completed, copies are distributed to co-workers
- Time is spent individually going through the work noting defects
- A meeting is held where the work is then discussed
- A list of defects requiring re-work is produced

Inspections - advantages of approach

- An effective way of removing superficial errors from a piece of software
- Motivates the software developer to produce better structured and self-descriptive code
- Spreads good programming practice
- enhances team-spirit
- *The main problem* maintaining the commitment of participants

‘Clean-room’ software development

Ideas associated with Harlan Mills at IBM

- Three separate teams:

Specification team – documents user requirements and usage profiles (how much use each function will have)

Development team – develops code but does not test it. Uses mathematical verification techniques

Certification team – tests code. Statistical model used to decide when to stop

Formal methods

- Use of mathematical notations such as VDM and Z to produce unambiguous specifications
- Can prove correctness of software mathematically (cf. geometric proofs of Pythagoras' theorem)
- Newer approach use Object Constraint Language (OCL) to add detail to UML models
- Aspiration is to be able to generate applications directly from UML+OCL without manual coding – Model Driven Architectures (MDA)

Quality plans

- quality standards and procedures should be documented in an organization’s *quality manual*
- for each separate project, the quality needs should be assessed
- select the level of quality assurance needed for the project and document in a *quality plan*

Typical contents of a quality plan

- scope of plan
- references to other documents
- quality management, including organization, tasks, and responsibilities
- documentation to be produced
- standards, practices and conventions
- reviews and audits

more contents of a quality plan

- testing
- problem reporting and corrective action
- tools, techniques, and methodologies
- code, media and supplier control
- records collection, maintenance and retention
- training
- risk management