ADVANCED TOPICS IN SOFTWARE ENGINEERING

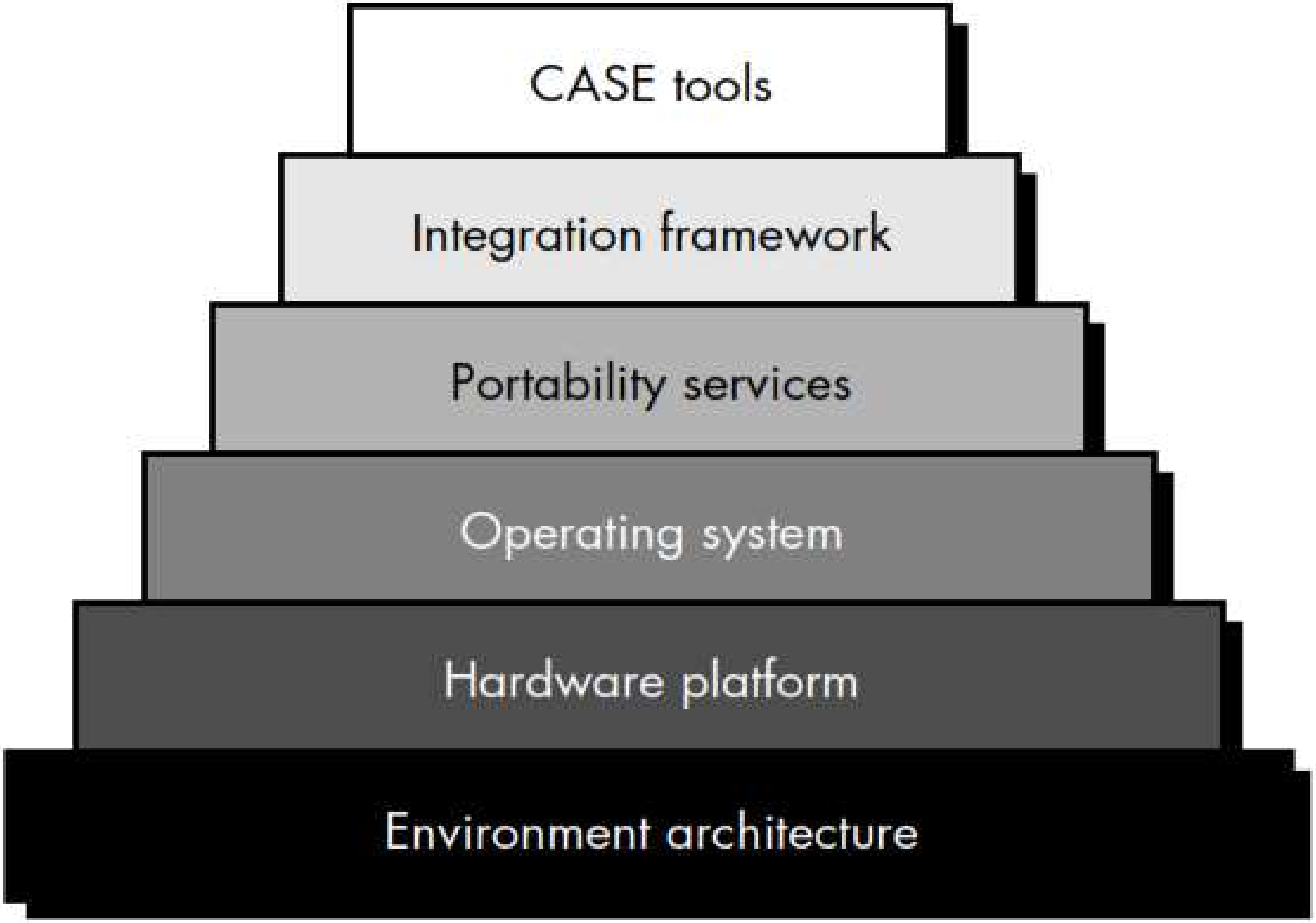
COMPUTER AIDED SOFTWARE ENGINEERING (CASE)

- CASE tools help in software development and maintenance.
- It is a mostly used in software industries.
- CASE tool is a general term:
  - Used to represent any form of automated support for software engineering or automates some software development activity.
- Some CASE tools assist in phase-related tasks:
  - specification, structured analysis, design, coding, testing, etc.
- Other tools help non-phase activities:
  - project management and configuration management.

CONT…

- CASE tools assist software engineering managers and practitioners in every activity associated with the software process.
- They automate project management activities, manage all work products produced throughout the process and assist engineers in their analysis, design, coding and test work.
- The main objective of the CASE is to increase productivity and help to produce better quality software at lower cost.

CASE TOOLS BUILDING BLOCKS



CONT…

- The environment architecture, composed of the hardware platform and system support (including networking software, database management and object management services).
- A set of portability services provides a bridge between CASE tools and their integration framework and the environment architecture.
- The integration framework is a collection of specialized programs that enables individual CASE tools to communicate with one another, to create a project database and to exhibit the same look and feel to the end-user.
- Portability services allow CASE tools and their integration framework to migrate across different hardware platforms and operating systems without adaptive maintenance.

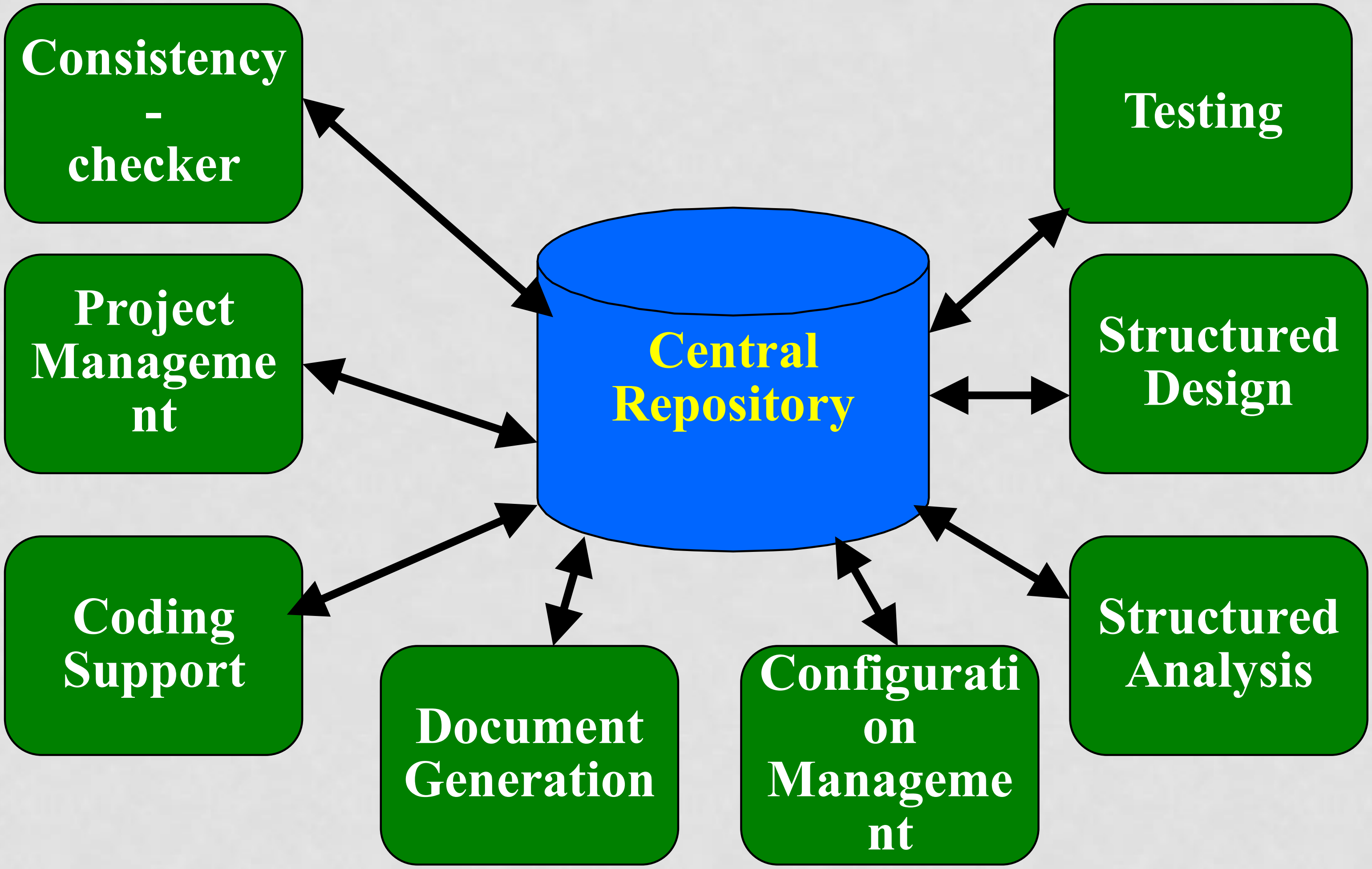
CASE ENVIRONMENT

- Individual CASE tools are useful but
  - true power of a tool set can be realized only when all CASE tools are integrated together into common framework or environment.
  - If CASE tools are not integrated, then the data generated by one tool is input to the other tools (additional effort of exporting data and importing data).
- Tools covering different stages of life cycle share information:
  - they should integrate through some central repository (store)

CONT…

- The central repository is the data dictionary:
  - contains definition of all composite and elementary data items.
  - through this repository all CASE tools share information.
- A CASE environment helps:
  - automate step-by-step methodologies.
- In contrast to CASE environment:
  - a programming environment denotes tools supporting only coding phase.

ARCHITECTURE OF CASE ENVIRONMENT



BENEFITS OF CASE

- A key benefit of using CASE environment:
  - cost saving through all developmental phases.
- Use of CASE tools leads to improvements in quality:
  - becomes easy to iterate through different software development phases.
  - chances of human error is reduced.
  - CASE tools help produce higher quality and consistent documents.
- Data relating to a software product are maintained in a central repository:
  - redundancy in the stored data is reduced.
  - chances of inconsistent documentation is reduced.

CONT…

- CASE tools take out most of the hard work from software engineers’ work:
  - engineers need not manually check balancing of the DFDs
  - easily draw diagrams and produce documentation, etc.
- CASE tools lead to cost saving in software maintenance effort:
  - traceability and consistency checks
  - systematic information capture during various development phases.

COMPONENT-BASED SOFTWARE ENGINEERING

- Component-based software engineering (CBSE) is a process that emphasizes the design and construction of computer-based systems using reusable software “components.”
- The process begins when a software team establishes requirements for the system.

ACTIVITIES

- Component qualification
  - System requirements and architecture define the components that will be required.
  - Reusable components are normally identified by the characteristics of their interfaces.
- Component adaptation
  - Software architecture represents design patterns that are composed of components (units of functionality), connections and coordination.
- Component composition
  - Architectural style again plays a key role in the way in which software components are integrated to form a working system.
- Component update
  - When systems are implemented with components, update is complicated by the imposition of a third party (i.e., the organization that developed the reusable component may be outside the immediate control of the software engineering organization).

VARIOUS COMPONENTS

- Component
  - a nontrivial, independent and replaceable part of a system that fulfills a clear function in the context of a well-defined architecture.
- Run-time software component
  - a dynamic bindable package of one or more programs managed as a unit and accessed through documented interfaces that can be discovered in run time.
- Software component
  - a unit of composition with contractually specified and explicit context dependencies only.
- Business component
  - The software implementation of an “autonomous” business concept or business process.

CONT…

- Qualified components
  - assessed by software engineers to ensure that not only functionality but performance, reliability, usability and other quality factors follow to the requirements of the system or product to be built.
- Adapted components
  - adapted to modify unwanted or undesirable characteristics.
- Updated components
  - replacing existing software as new versions of components become available.

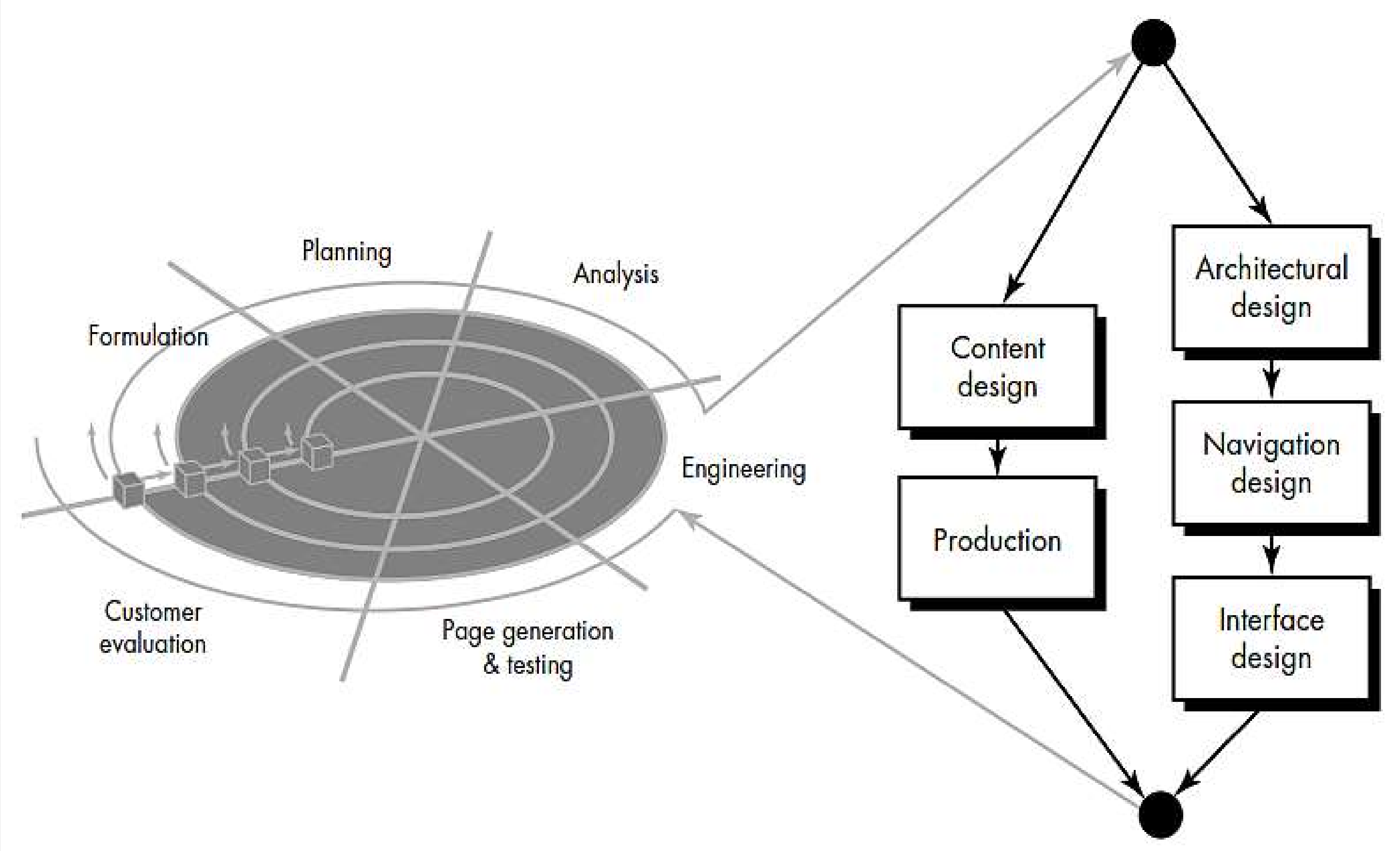
WEB ENGINEERING

- Web engineering is the process used to create high-quality Web Apps.
- Web engineering is not a perfect clone of software engineering, but it borrows many of software engineering's fundamental concepts and principles, emphasizing the same technical and management activities.
- It is concerned with the establishment and use of scientific, engineering and management principles and systematic approaches to the successful development and maintenance of high quality Web-based systems and applications.

FRAME WORK FOR WEB ENGINEERING

- As WebApps develop from static, content-directed information sources to dynamic, user-directed application environments.
- To accomplish this, it is necessary to develop a WebE framework that encompasses an effective process model, populated by framework activities and engineering tasks.

THE WEBE PROCESS MODEL



CONT…

- The WebE process begins with a formulation an activity that identifies the goals and objectives of the WebApp and establishes the scope for the first increment.
- Planning estimates overall project cost, evaluates risks associated with the development effort and defines rough development schedule for the initial WebApp increment.
- Schedule for subsequent increments.
- Analysis establishes technical requirements for the WebApp and identifies the content items that will be incorporated.
- Requirements for graphic design are also defined.
- Content design and production are tasks performed by nontechnical members of the WebE team.

CONT…

- The intent of these tasks is to design, produce and/or acquire all text, graphics, audio and video content that are to become integrated into the WebApp.
- Page generation is a construction activity that makes heavy use of automated tools for WebApp creation.
- Testing exercises
  - WebApp navigation
  - attempts to uncover errors in scripts and forms
  - helps ensure that the WebApp will operate correctly in different environments (e.g., with different browsers).
- Each increment produced as part of the process is reviewed during customer evaluation.