

Component-based Development Process and Component Lifecycle

Ivica Crnkovic
Mälardalen University,
Department of Computer Engineering
Sweden
ivica.crnkovic@mdh.se
<http://www.idt.mdh.se/~icc>

Software Development processes

- What determines which development process model to use?
 - Type of products/products (requirements from customers)
 - Type of project
 - Availability& requirements of stakeholders
 - Type of organization
 - **Technology used**
 -

Software development process adaptation

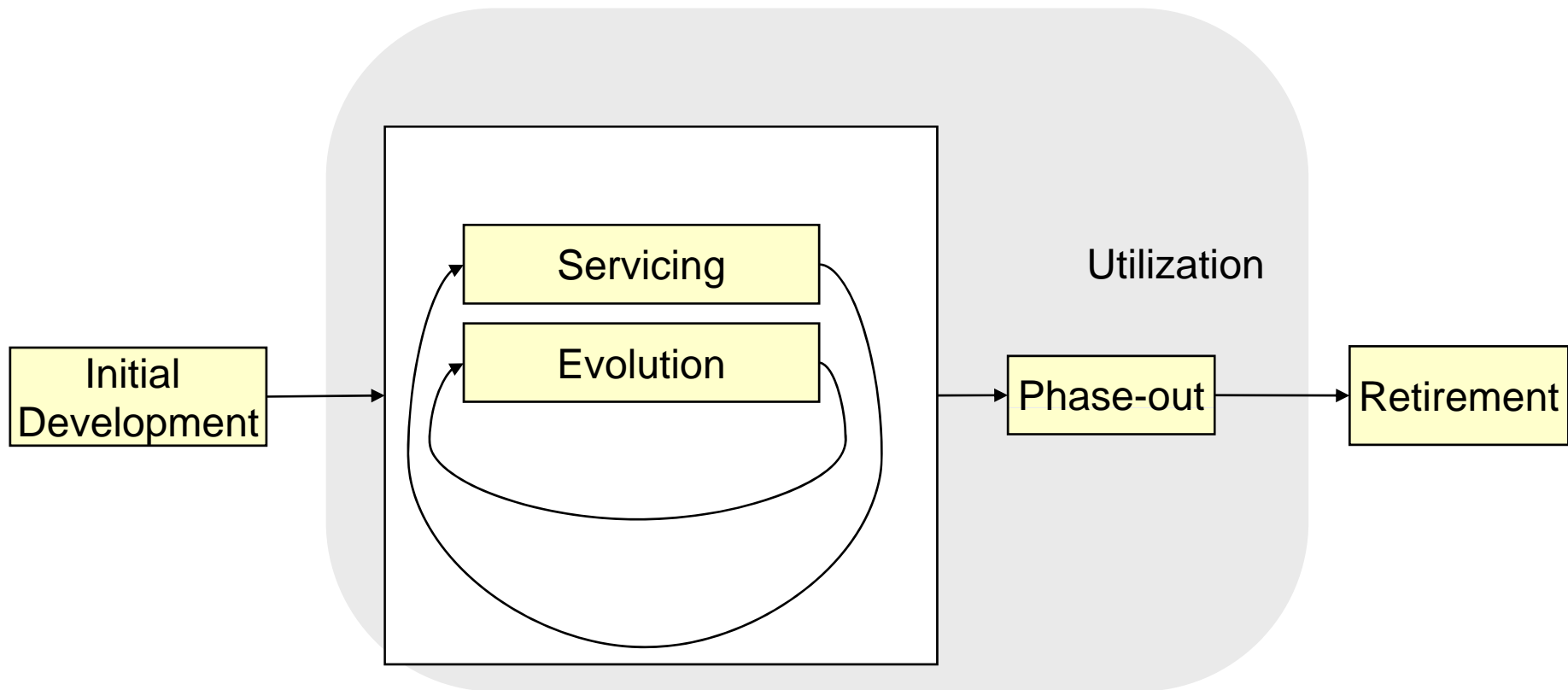
- Software development processes are usually of generic type
 - Usually requires adaptations
 - Often a software development process is a combination of several models
 - There is difference between theory and practice
 - Practice is often more complex
 - Practice is not perfect

Lifecycle Process Models for products



Generic Product Lifecycle

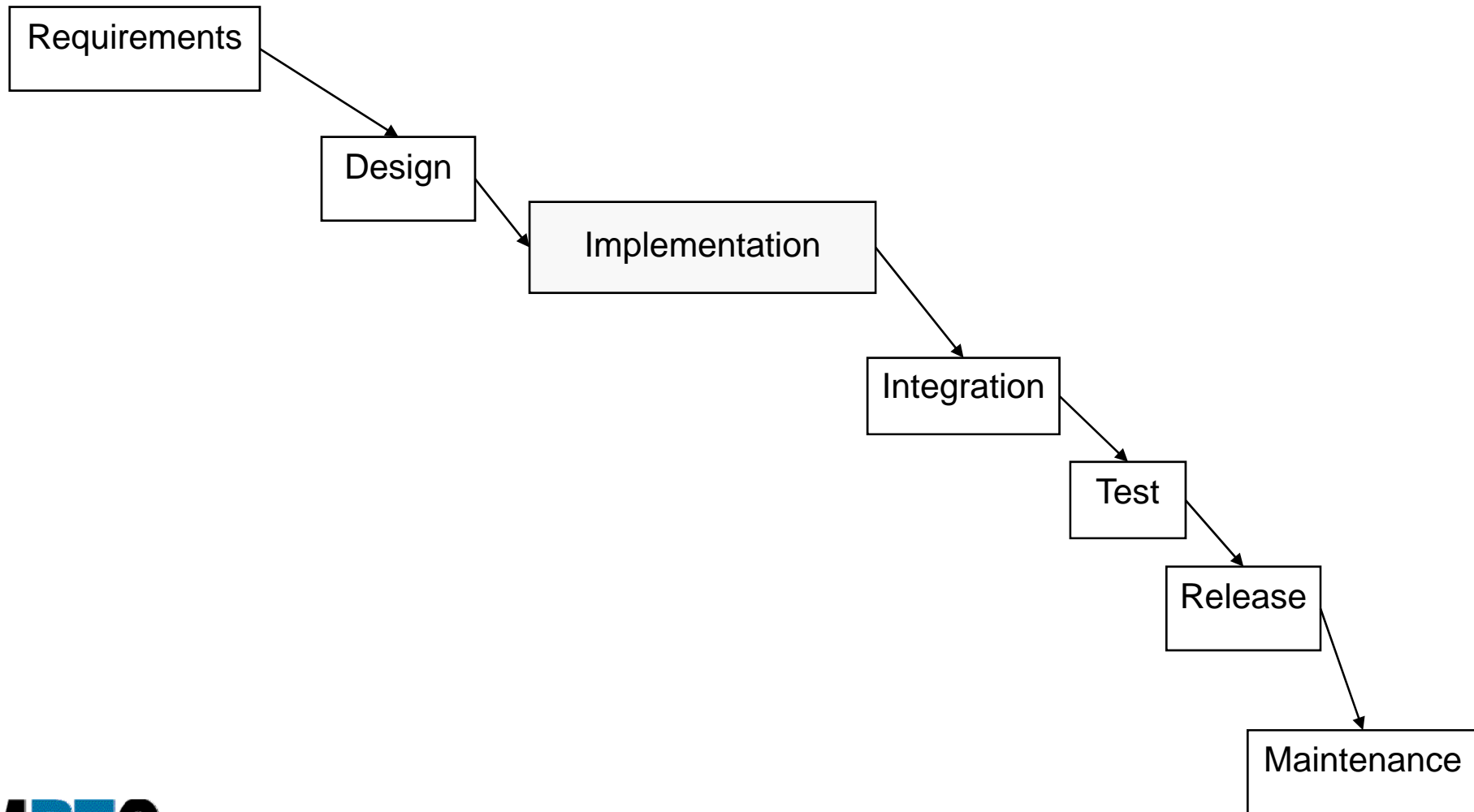
Lifecycle Process Models for software products

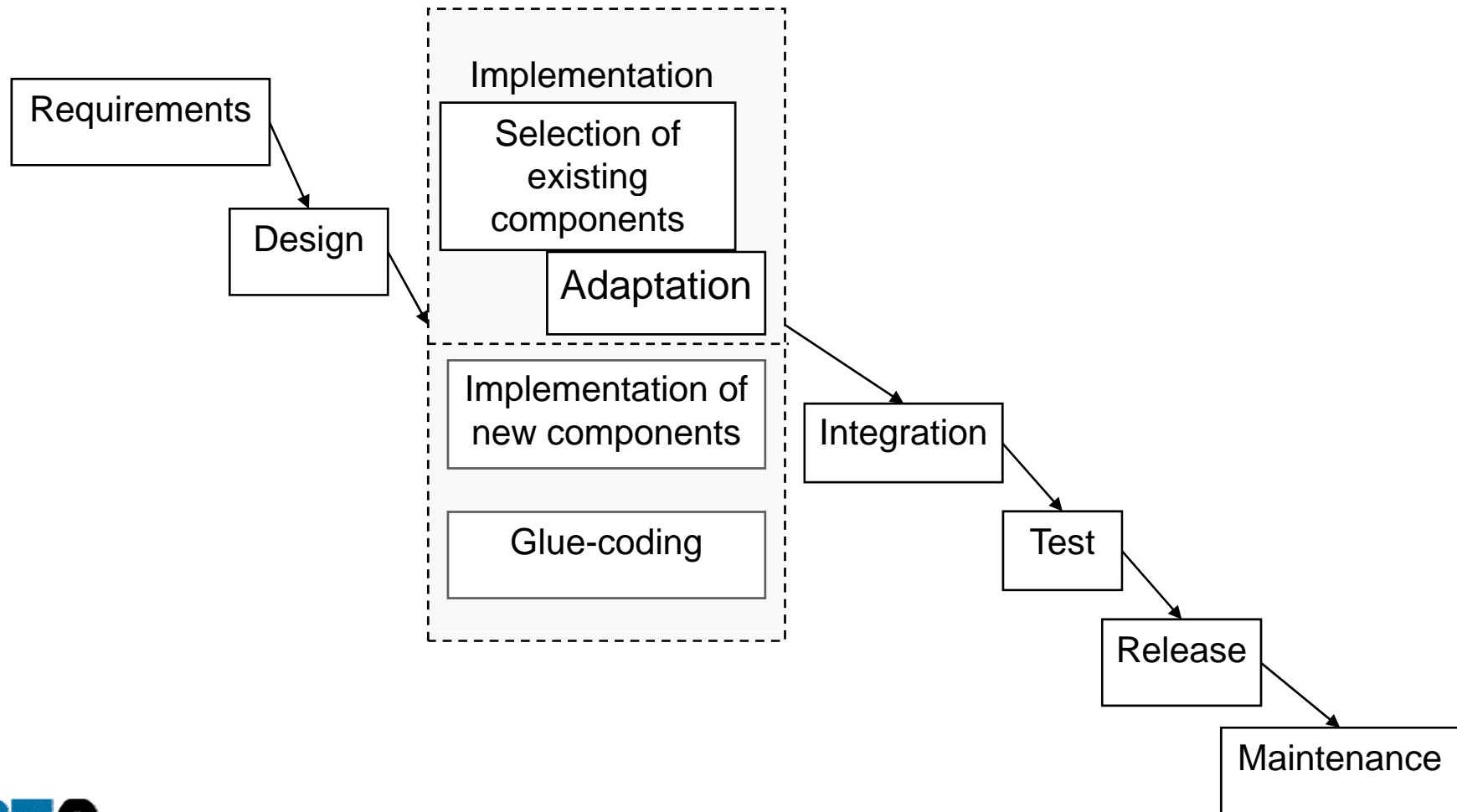


Component-based approach process characteristics

- **Separation of the development process.**
 - The development processes of component-based systems
 - development processes of the components.
- **A new process: Component Assessment.**
 - Finding and evaluating the components.
- **Changes in the activities in the development processes.**
 - system-level process the emphasis will be on finding the proper components and verifying them,
 - component-level process, design for reuse will be the main concern.

Waterfall model

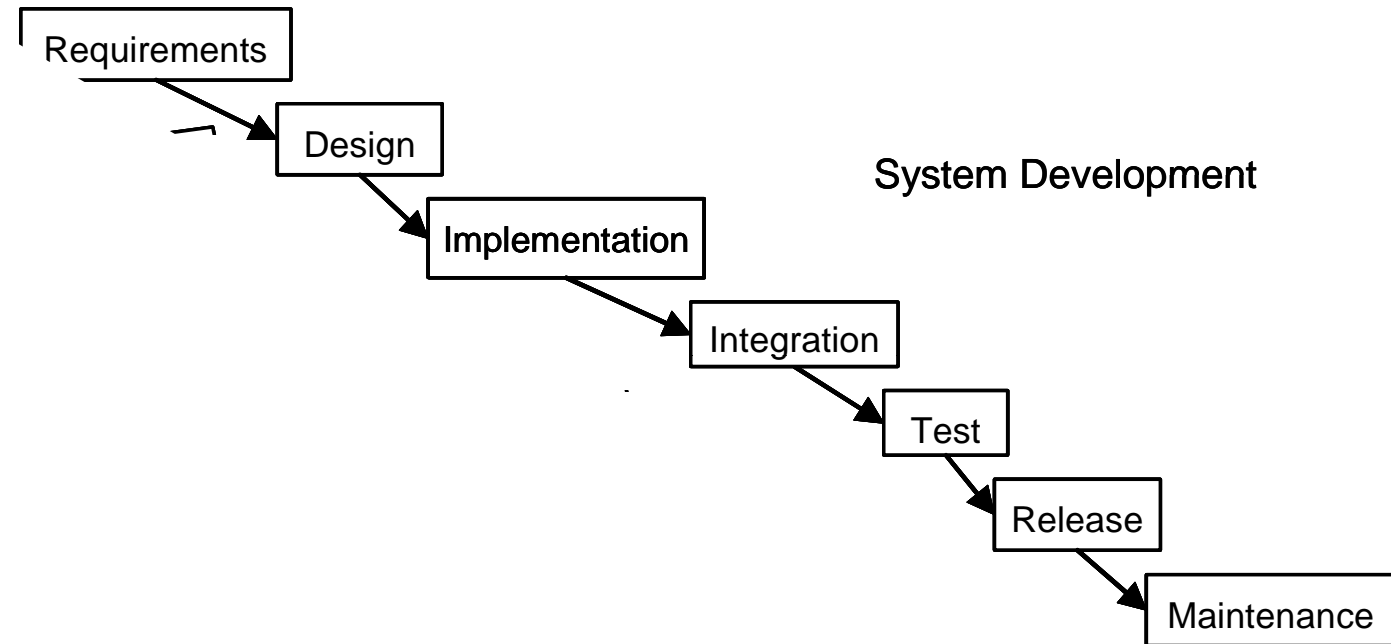




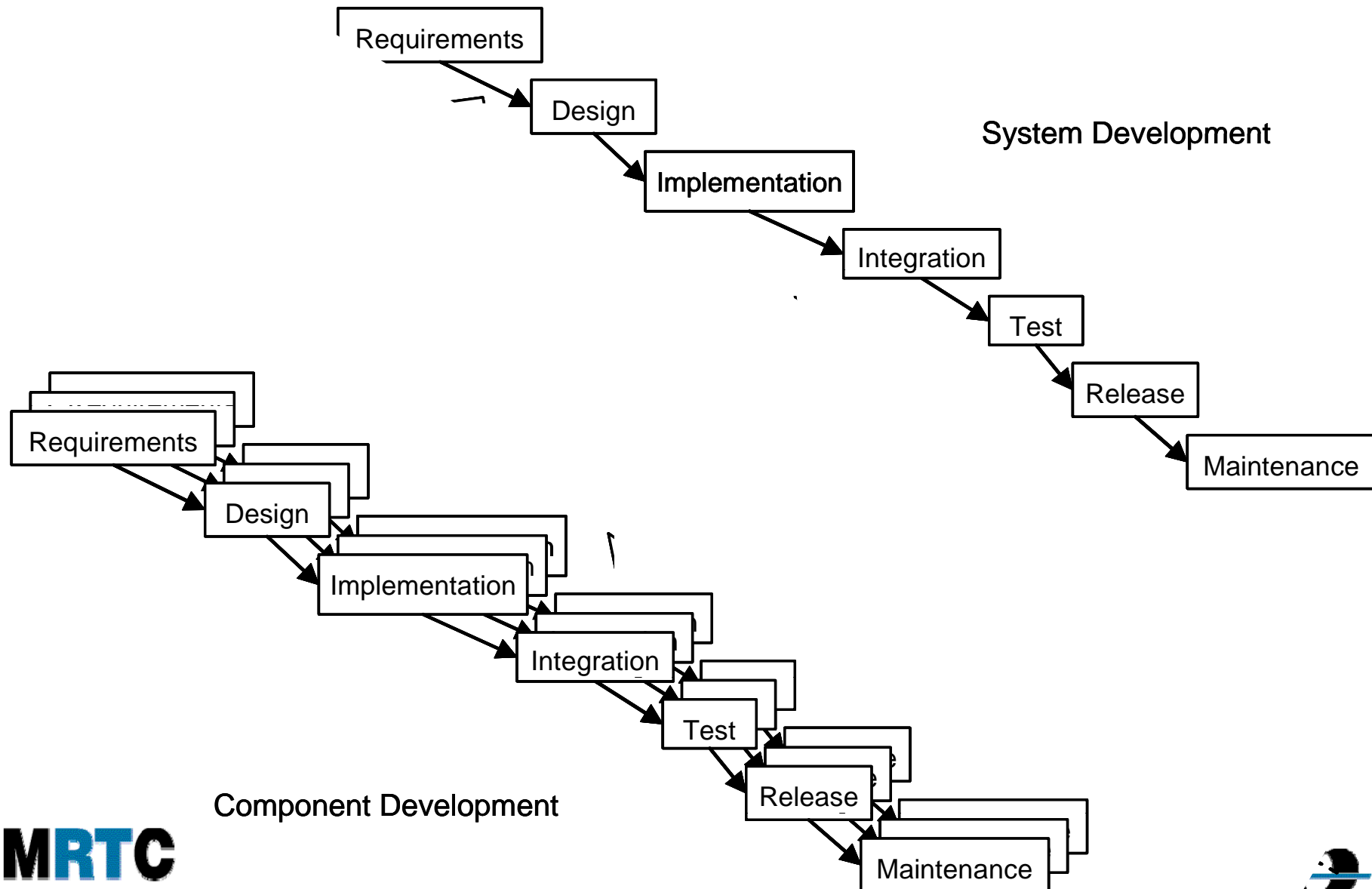
A simplified and an idealized process

- Assumption of the model
 - components selected and used are sufficiently close to the units identified in the design process
- Further, the figure shows only the process related to the system development – not to the supporting processes

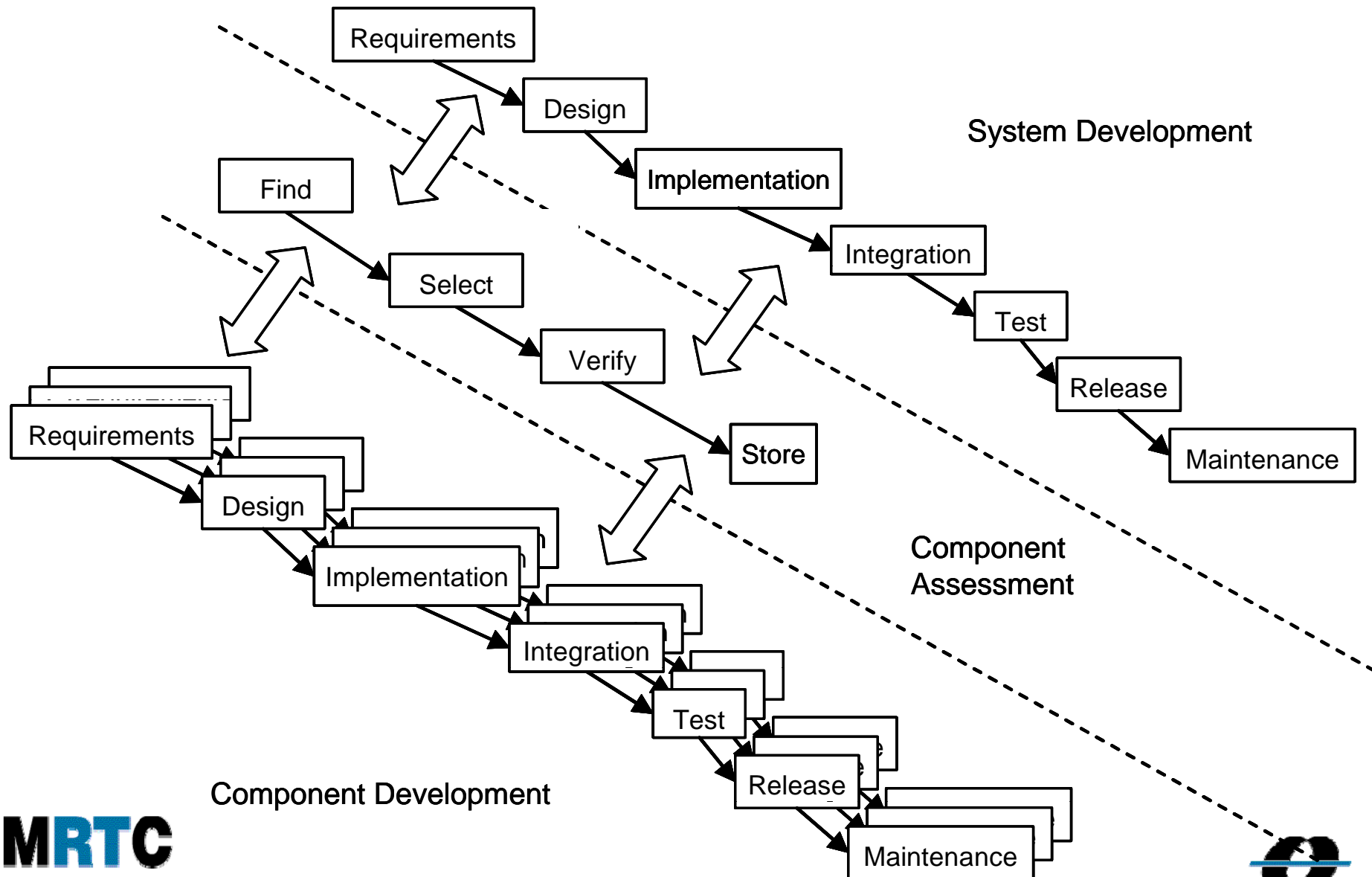
CB System Development...



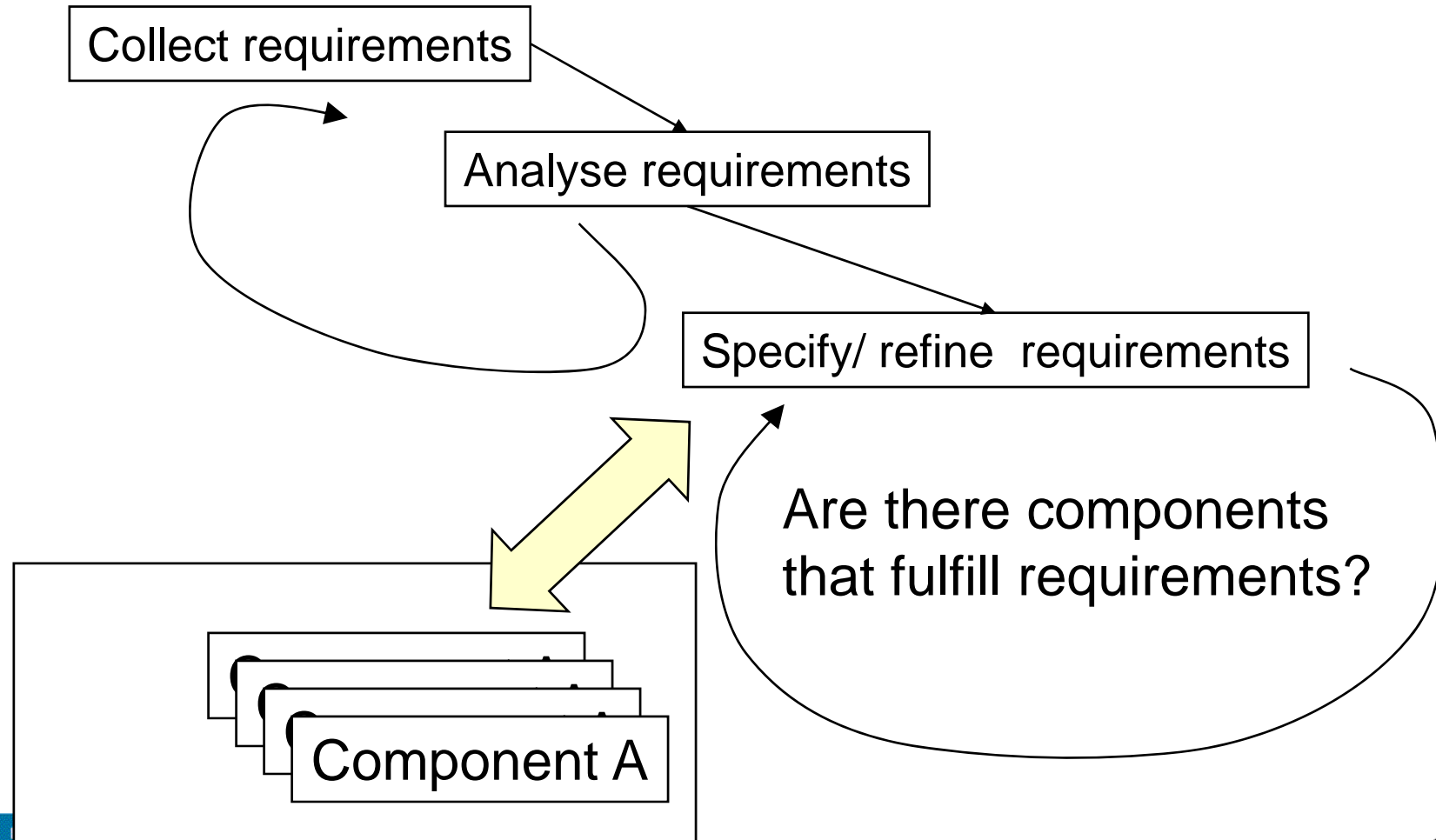
CF system and Components Development



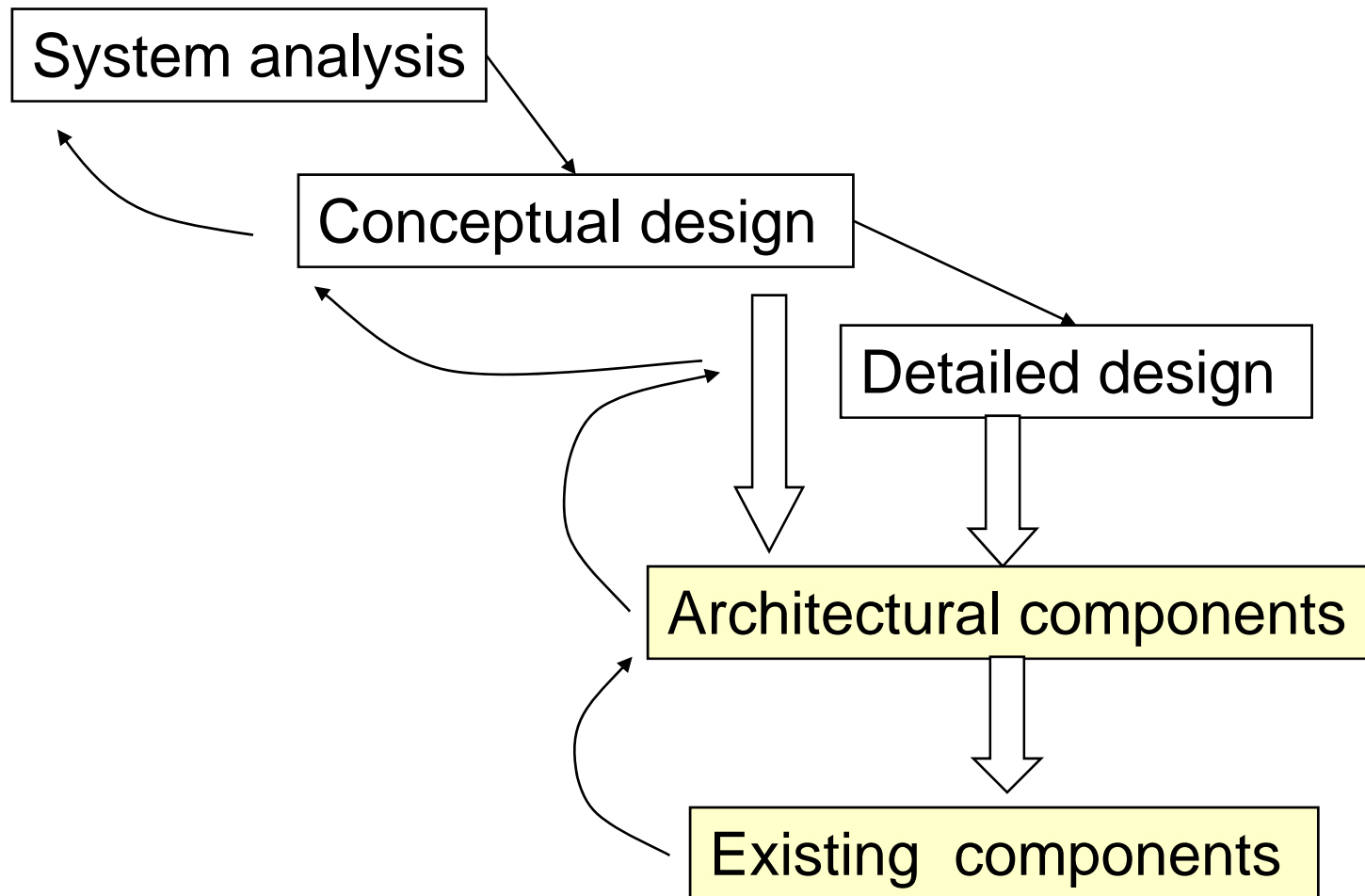
The complete process



System Requirements Phase

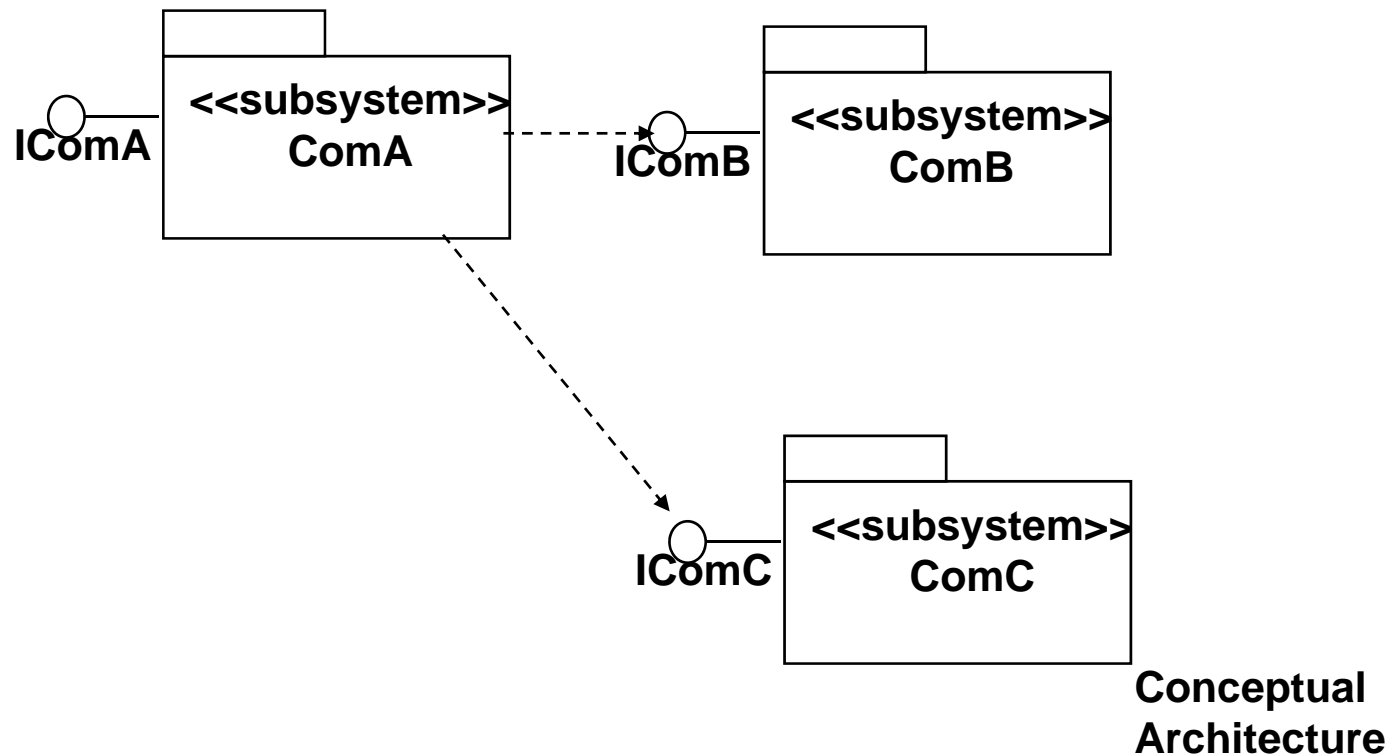


System and Analysis & Design Phase



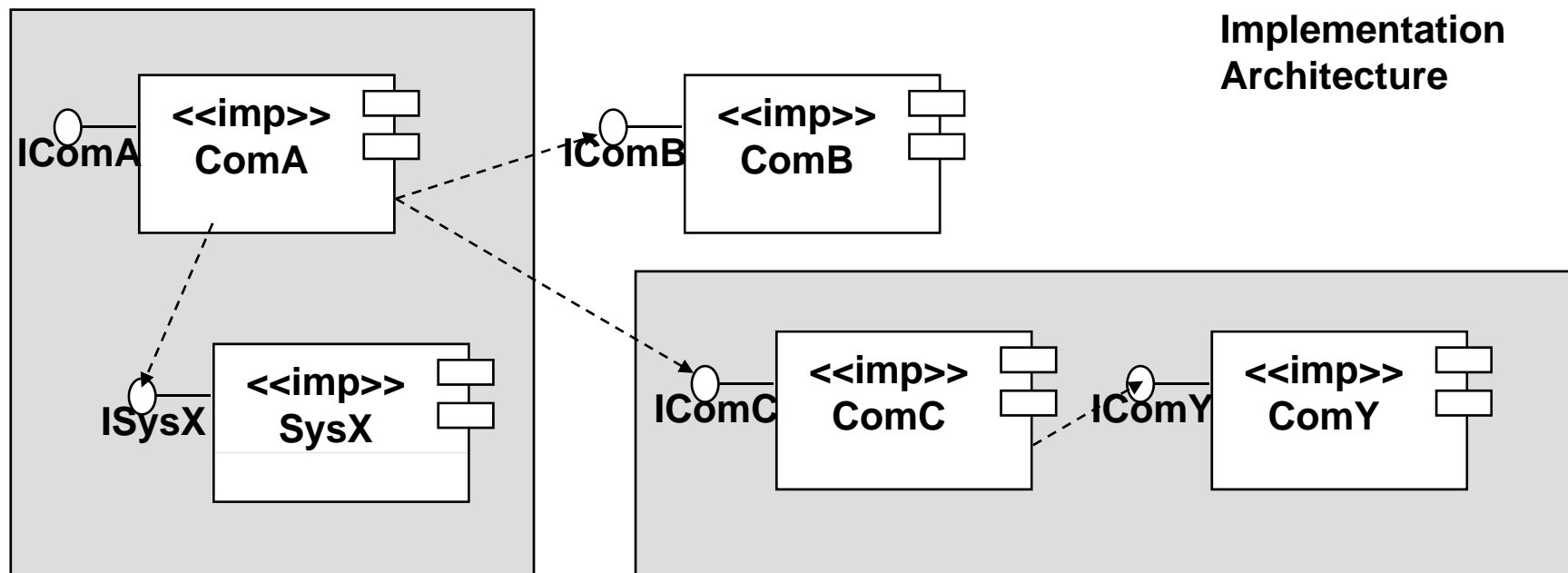
Different architecture view in different phases

- Phase I
 - System architecture - Decomposition of the system



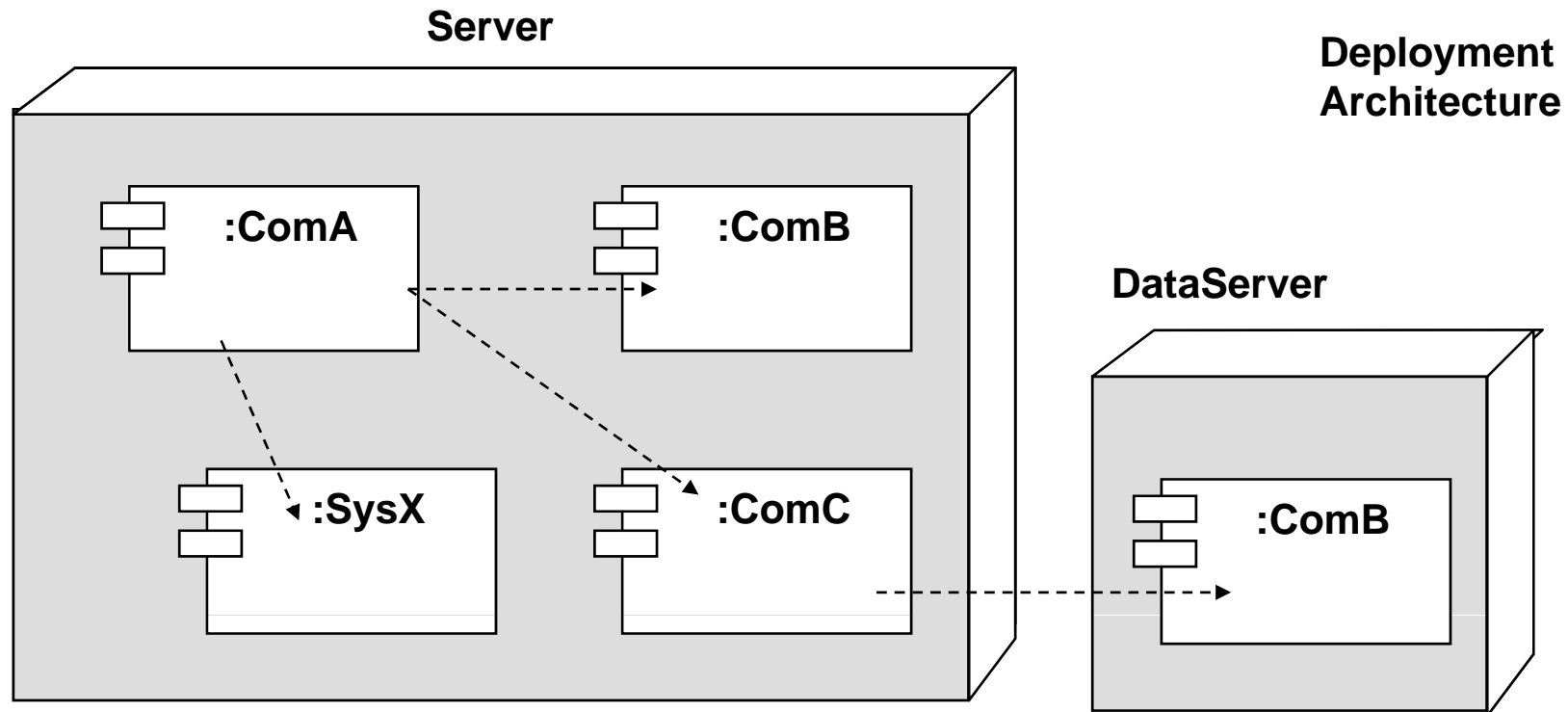
System Design – Phase 2

- Implementation Architecture - Component Identification

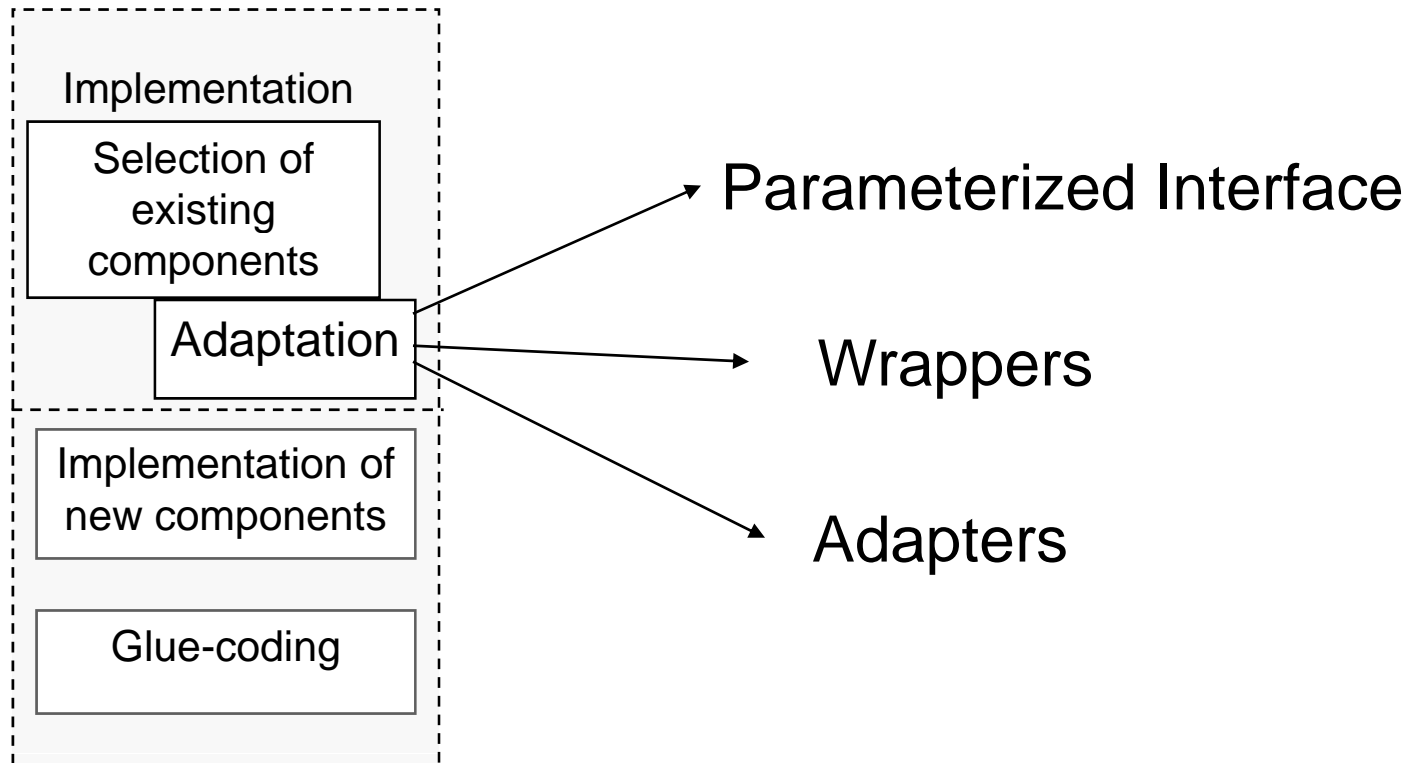


System Design – Phase 3

- Deployment architecture



System Implementation Phase

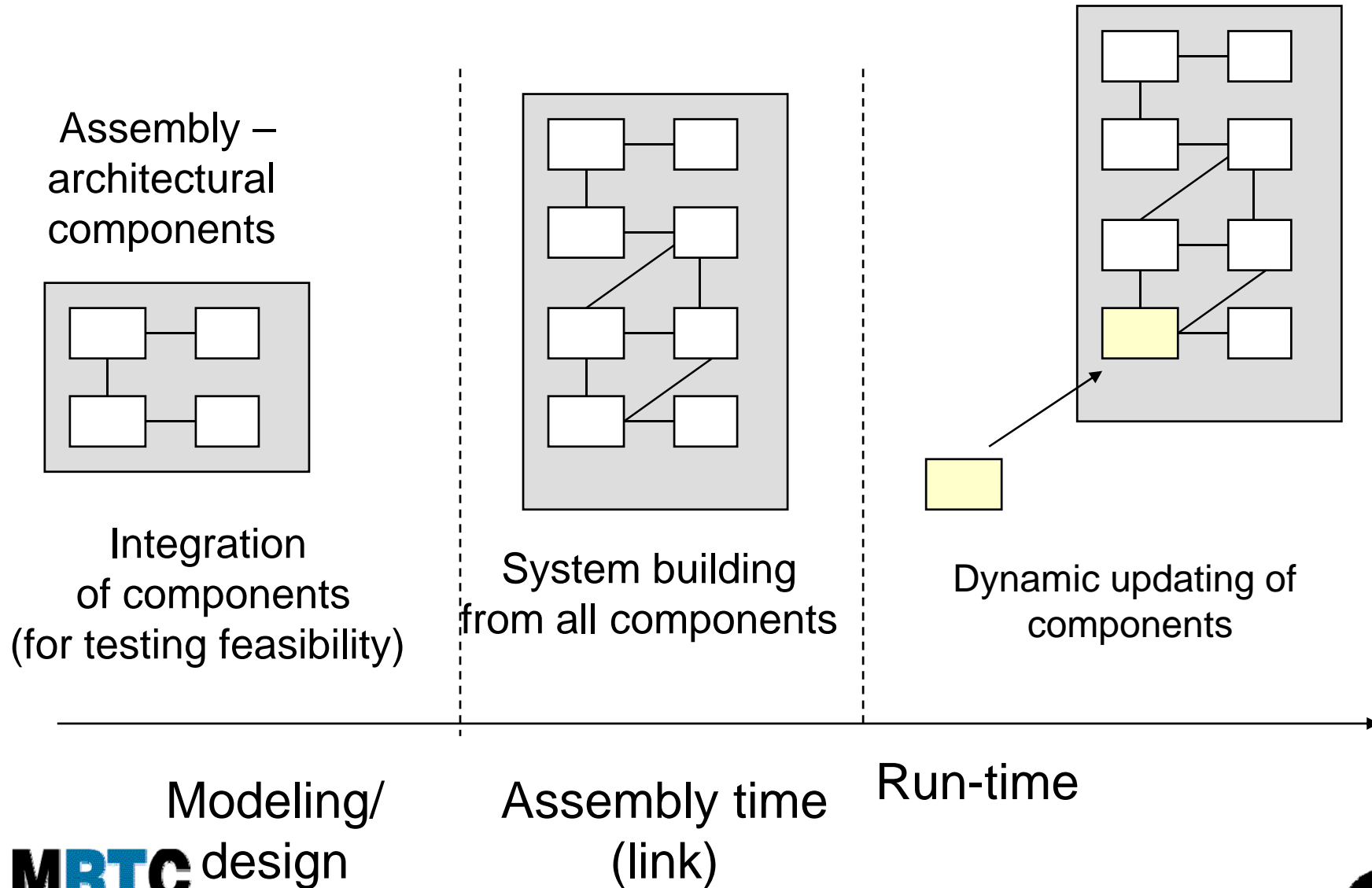


- **Parameterized Interface.** Parameterized interface makes it possible to change the component properties by specifying parameters that are the parts of the component interface.
- **Wrapper.** A wrapper is a special type of a glue-code that encapsulates a component and provides a new interface that either restrict or extend the original interface, or to add or ensure particular properties.
- **Adapter.** An adapter is a glue code that modifies ('adapts') the component interface to make it compatible with the interface of another component.

System Integration Process

- Putting components together
- Integration components into the system (component) framework
- Integration can happen in different phase of product's lifecycle

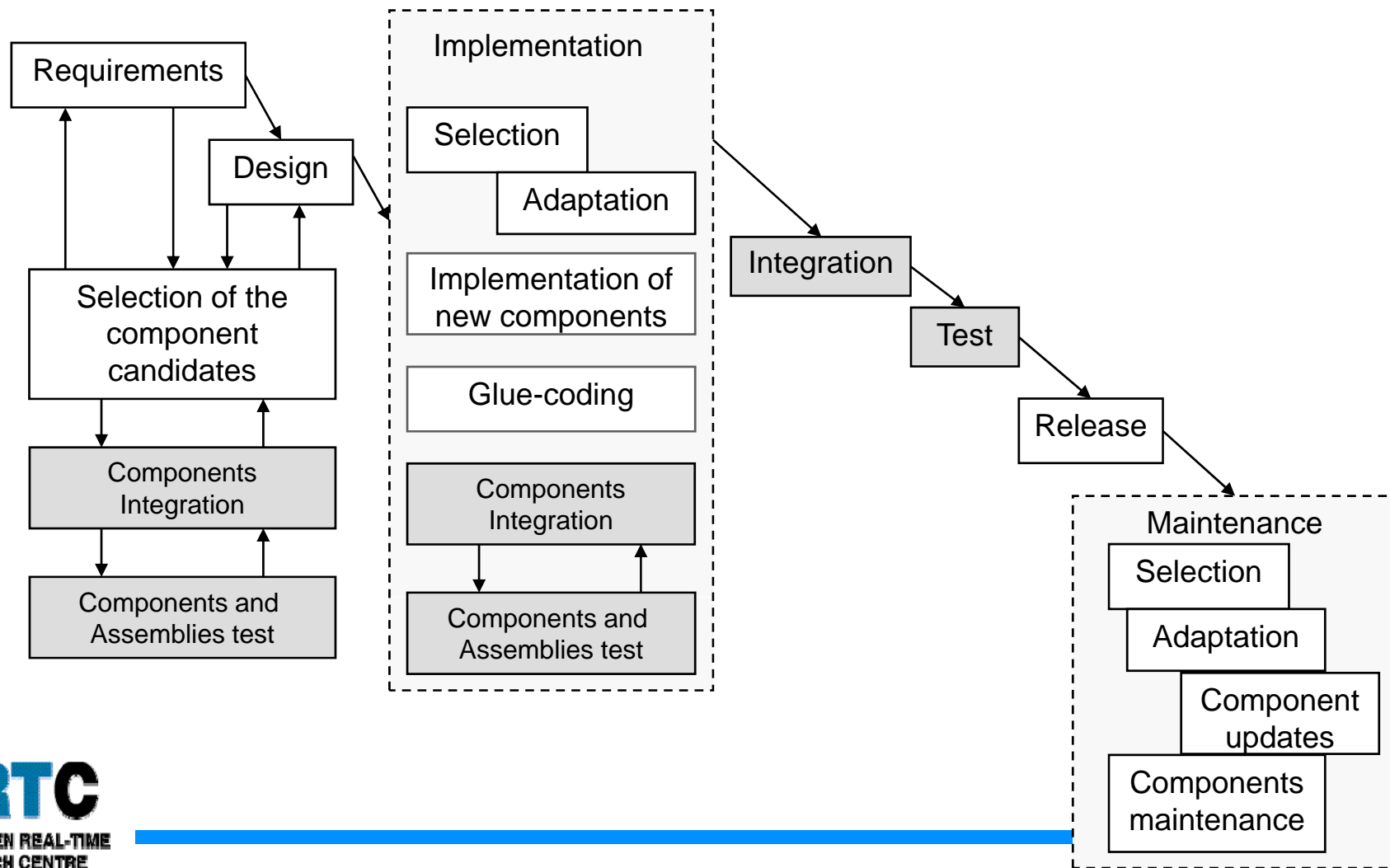
Integration in different phases



Test Phase

- System is being verified against the system specification
- In the waterfall model the test is performed after the system integrations,
- In CBD
 - Tests performed for isolated components
 - Tests of assemblies
 - Test of the system
- Tests are present in all phases!.

Integration and test in different phases of the CBD process

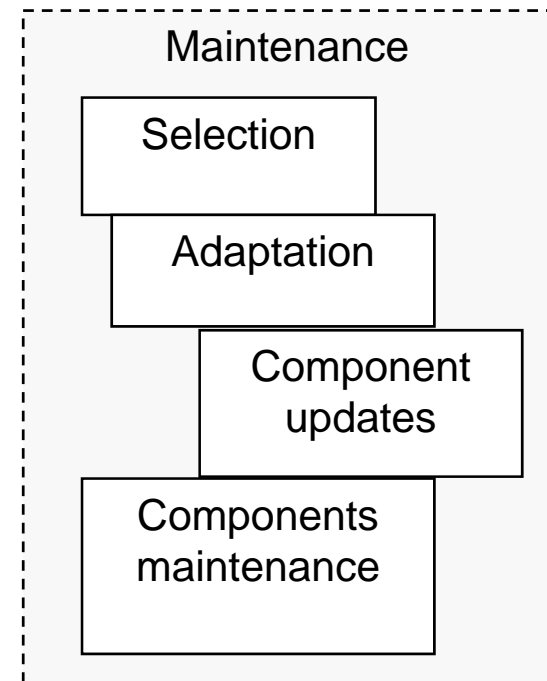


Release Phase

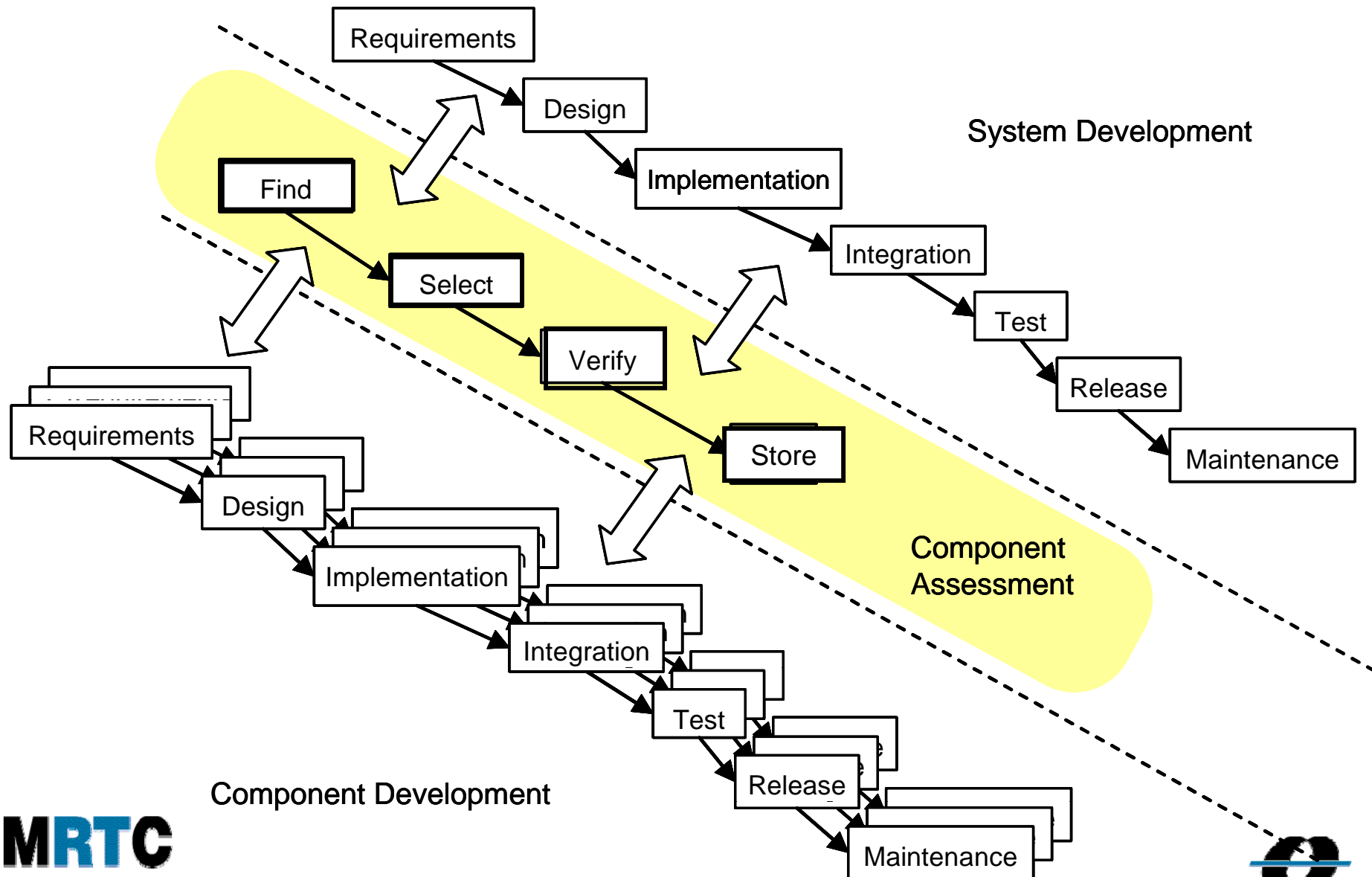
- packaging of the software in forms suitable for delivery and installation.
- The CBD release phase is not significantly different from a “classical” integration.

System Maintenance Phase

- The approach of CBD is to provide maintenance by replacing old components by new components or by adding new components into the systems.
- The paradigm of the maintenance process is similar to this for the development:
 - Find a proper component, test it, adopt it if necessary, and integrate it into the system



Component assessment process



Component assessment process

- CBD characteristics
 - Instead of developing find the components!
- Activates:
 - Find components (often a set of components must be found) that
 - Verify that the component(s) indeed provide the desired (or almost desired) functionality,
 - Verify that the components can successfully be integrated with other components.
 - (The consequence can be that not the best components can be selected, but the components that fit together).

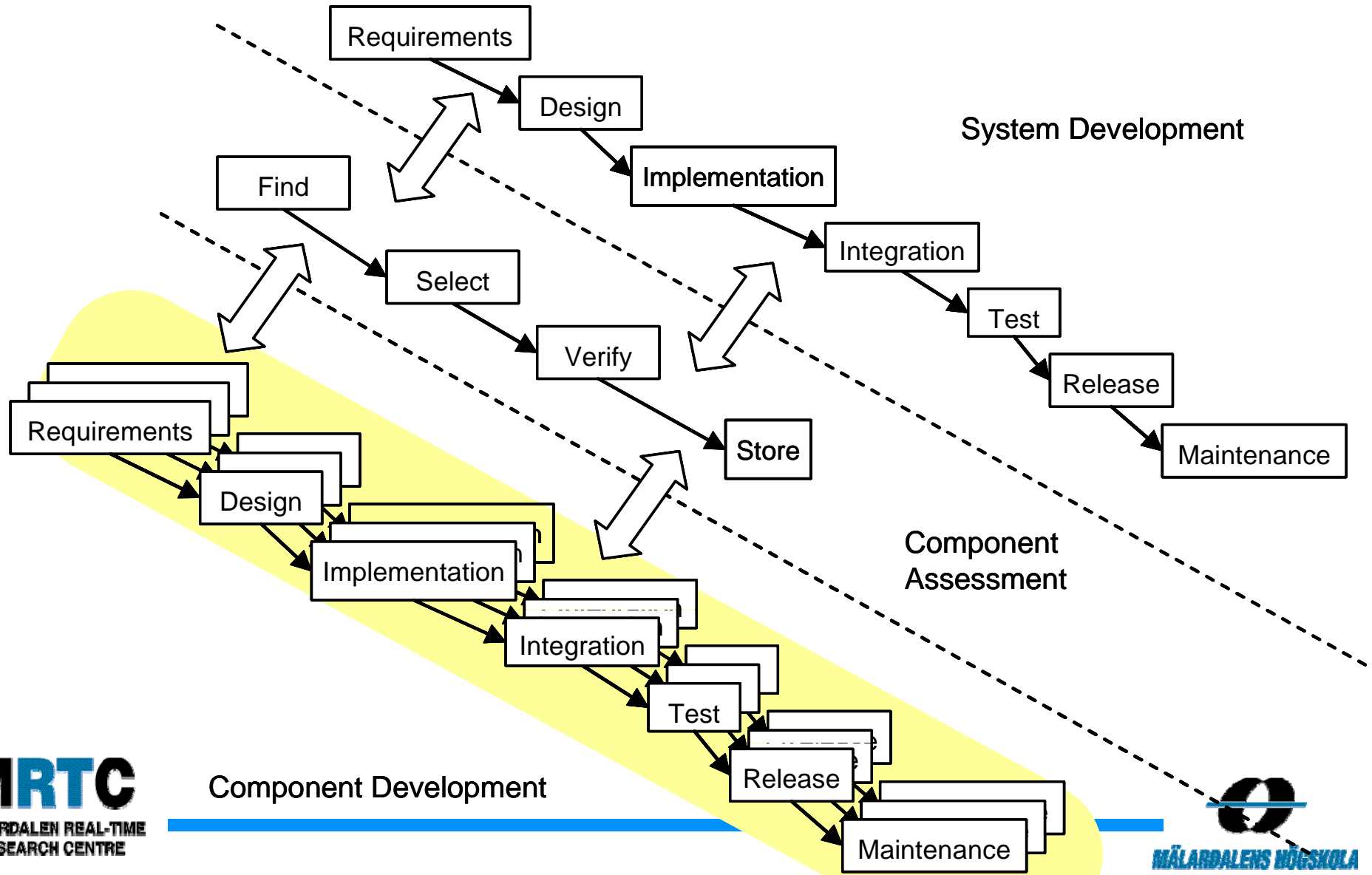
A generic assessment process activities

- **Find** – From an “infinite” component space find the components that might provide the required functionality.
-
- **Select** –Between the components candidates found, select a component that is most suitable for given requirements and constraints.
- **Verify** –
 - Test functional and certain extra-functional properties of a component in isolation.
 - Test the component in combination with other components integrated in an assembly.
- **Store** –
 - store the selected components in a component repository.
 - Store additional specification (metadata)
 - measured results of component performance,
 - known problems,
 - the tests and tests results and similar

A assessment process activities

- **Activities**
 - Find
 - Select
 - Verify
 - Store
- **The concrete activities dependent of type of component-based development process**
 - Architecture-driven component development
 - Product-line development
 - COTS-based development.

Component development process



Component development process - specifics

- Components are built as reusable units
- Components are built for future systems
- Consequences
 - There is greater difficulty in managing requirements;
 - Greater efforts are needed to develop reusable units;
 - Greater efforts are needed for providing component specifications and additional material that help developers/consumers of the components.

- There is greater difficulty in managing requirements;
- Greater efforts are needed to develop reusable units;
- Greater efforts are needed for providing component specifications and additional material that help developers/consumers of the components.

Requirements Phase

- A combination of a top-down and bottom-up process.
 - The requirements elicitation should be the result of the requirements specification on the system level.
 - Requirements for general types of functions/services
 - Reusability should be addressed explicitly.

Different architectural approaches in CBD

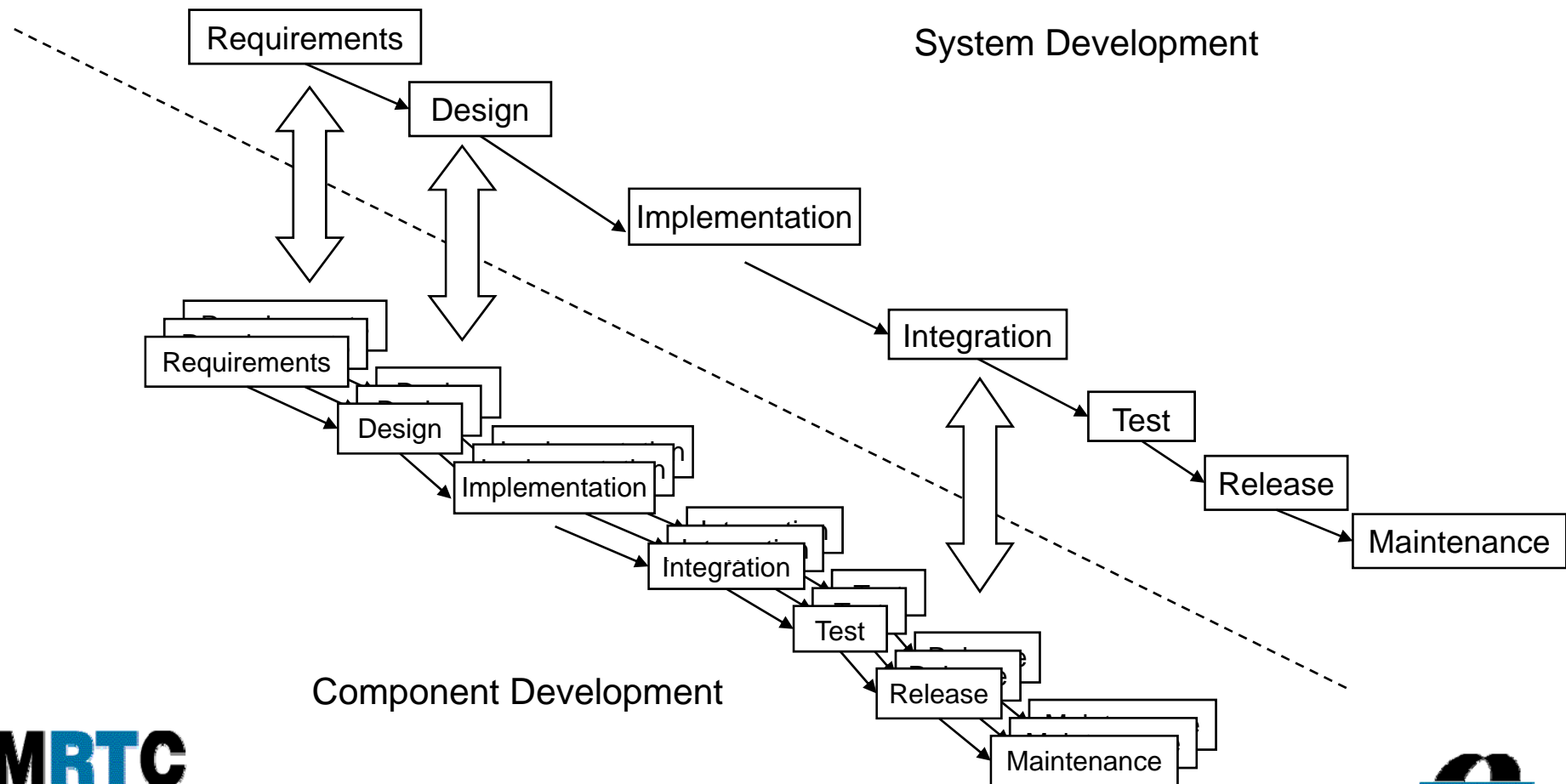
- Architecture-driven component development
- Product-line development
- COTS-based development

Architect-driven component development

- Top-down approach
 - components are identified as architectural elements and as a means to achieve a good design.
 - Components are not primarily developed for reuse,
 - Component-based technologies are used, because of easier implementations, in getting already existing services provided from the component technology.
 - the main characteristic of these components is composability,
 - No emphasis on while reusability
 - No emphasis time-to-market issues

Architect-driven component development

- The parallel development processes are reduced to two semi-parallel processes



A product line is:*

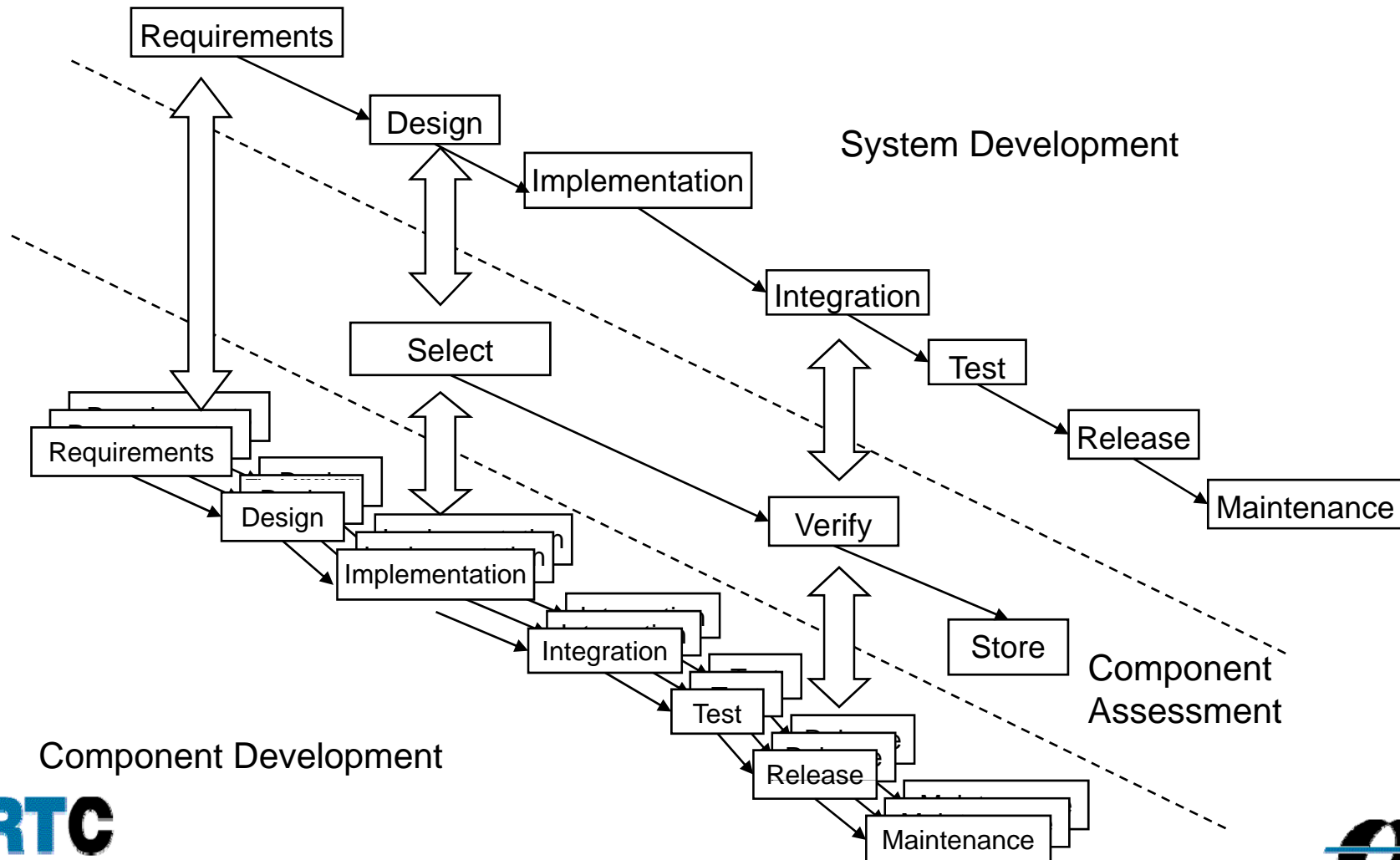


* From Rob Van Ommering/Philips

Product-line development

- GOAL
 - to enable efficient development of many variants of product, or family of products
 - to achieve a large commercial diversity (i.e. producing many variants and many models of products) with a minimal technical diversity at minimal costs [COPA].
 - They are heavily architecture-driven, as the architectural solution should provide the most important characteristics of the systems.
 - component-based approach enables reuse of components, and efficient integration process.
 - composability, reusability and time-to-market are equally important.
 - characteristic:
 - The component model must comply with the pre-defined reference architecture.
 - parallelism of component development process and product development process

Product-line development



COTS-based development

- COTS - commercial off the shelf
- component development process completely separately developed from system development.
- The strongest concern
 - time.-to-market from the component user point of view,
 - reusability from the component developer point of view.
- Main characteristics
 - instant value of new functionality
 - Challenges in composability
 - Often COTS components don't comply with a component model,
 - the semantics of the components are not specified
 - different properties of the components are not properly and adequately documented.

Case Study

- Philips Consumer Electronics (TV, Video, DVD)
 - Moving from a hardware local development to software & hardware global development
- Requirements
 - New products (product models, variants, etc.) must be delivered each 6 months

Experience

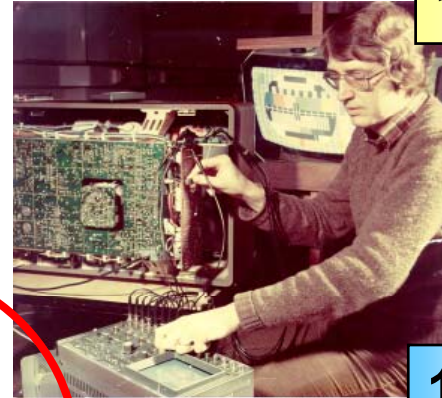
“hardware-like” componentization of software made it possible to make the transformation

The domain...

1965



1979



1 kB

2000



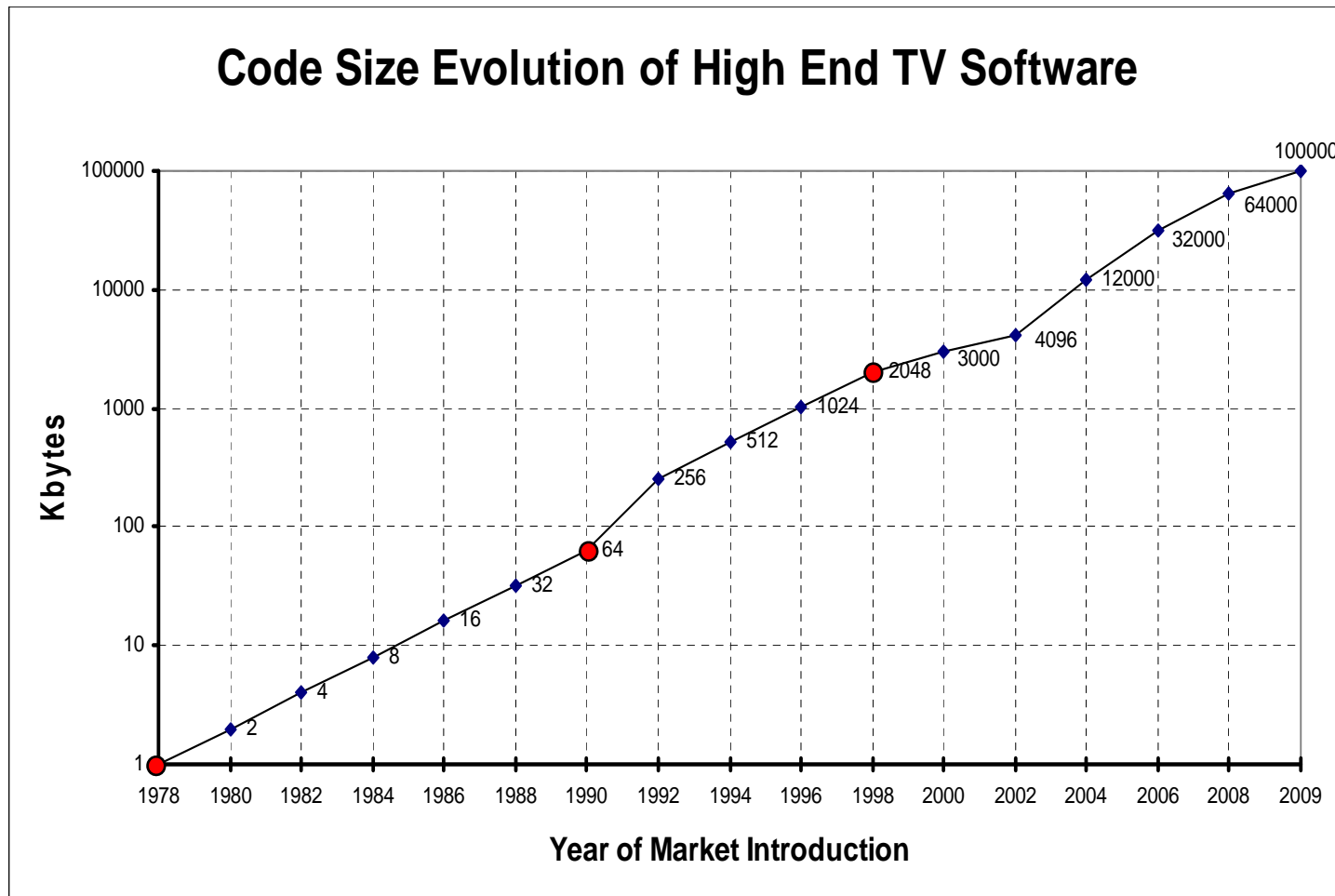
2 MB

1990



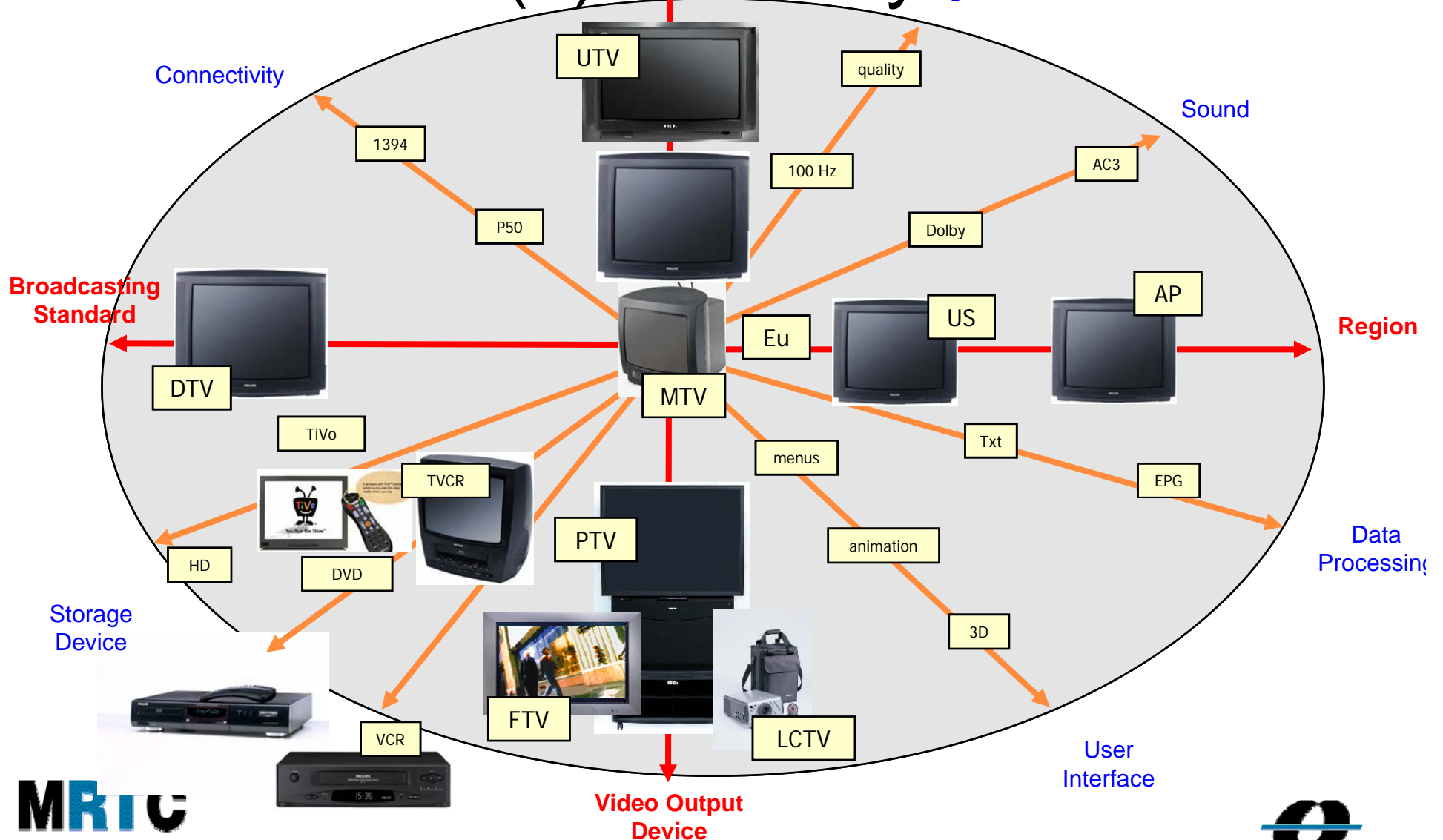
64 kB

(1) Complexity



* From Rob Van Ommering/Philips

(2) Diversity



(3) Lead Time

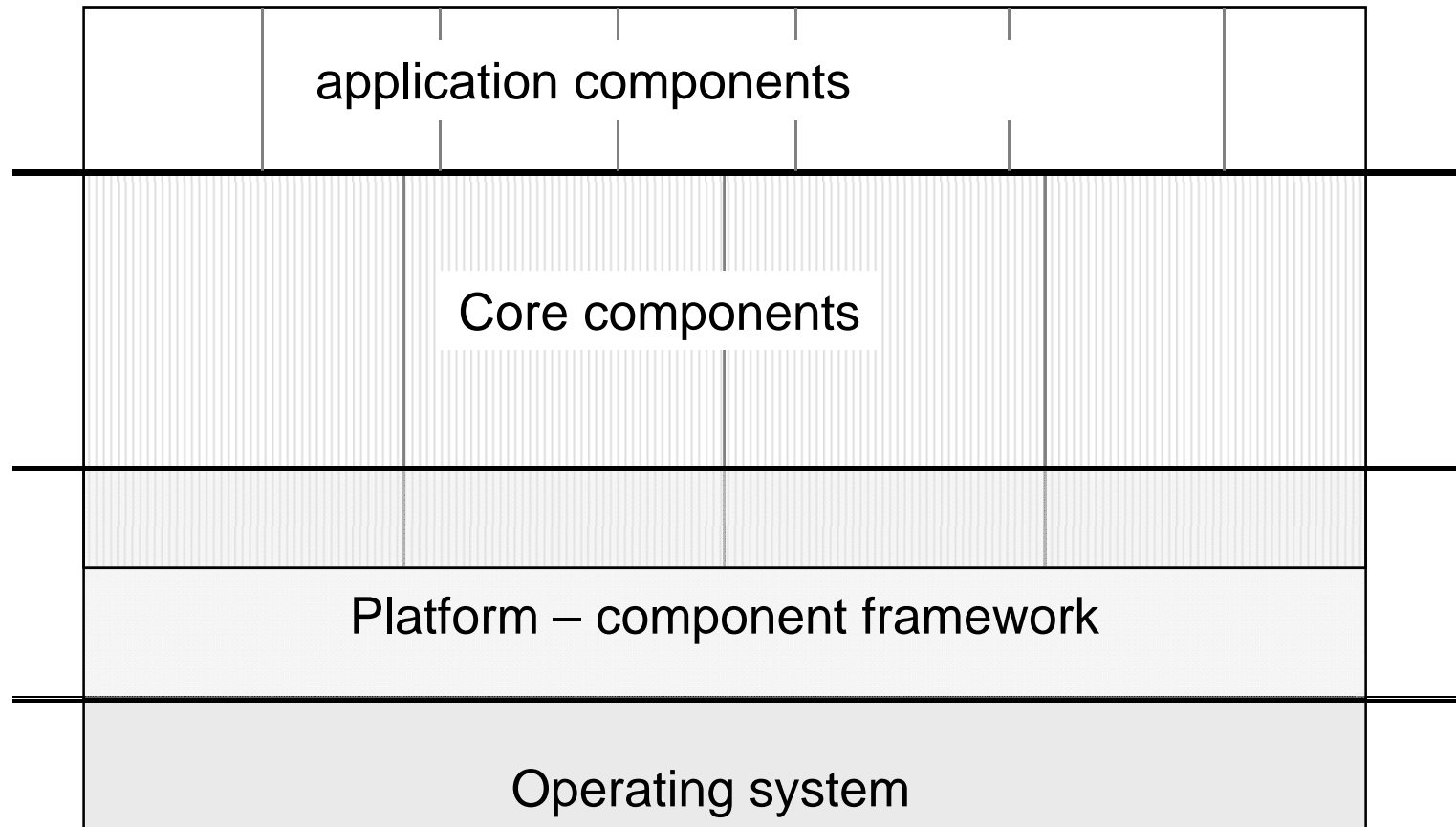
Was:

- Yearly cycle of product introduction
 - Christmas
 - World championship

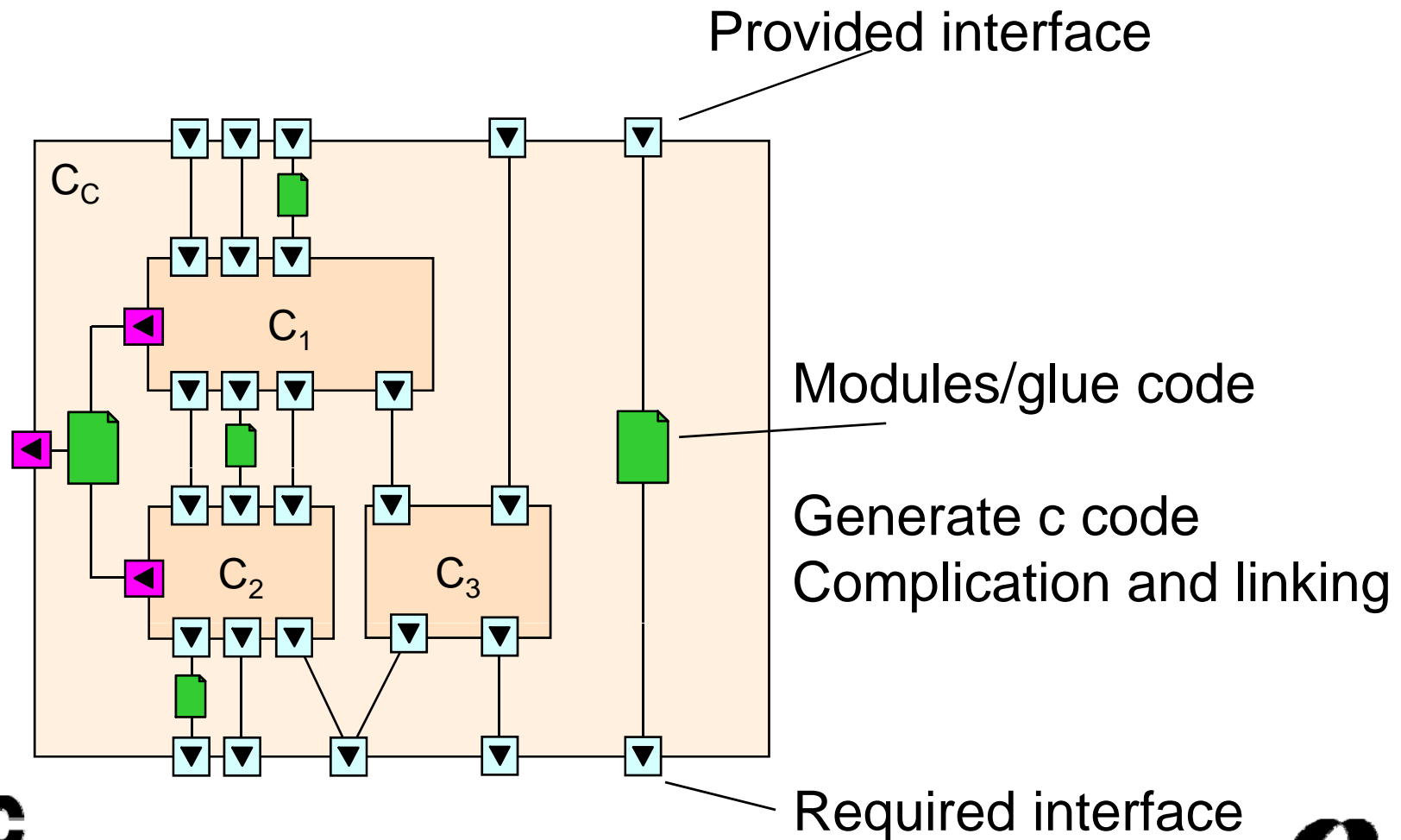
Is:

- Decreasing to 6 (or even 3) months
 - Otherwise loose shelf space in shop

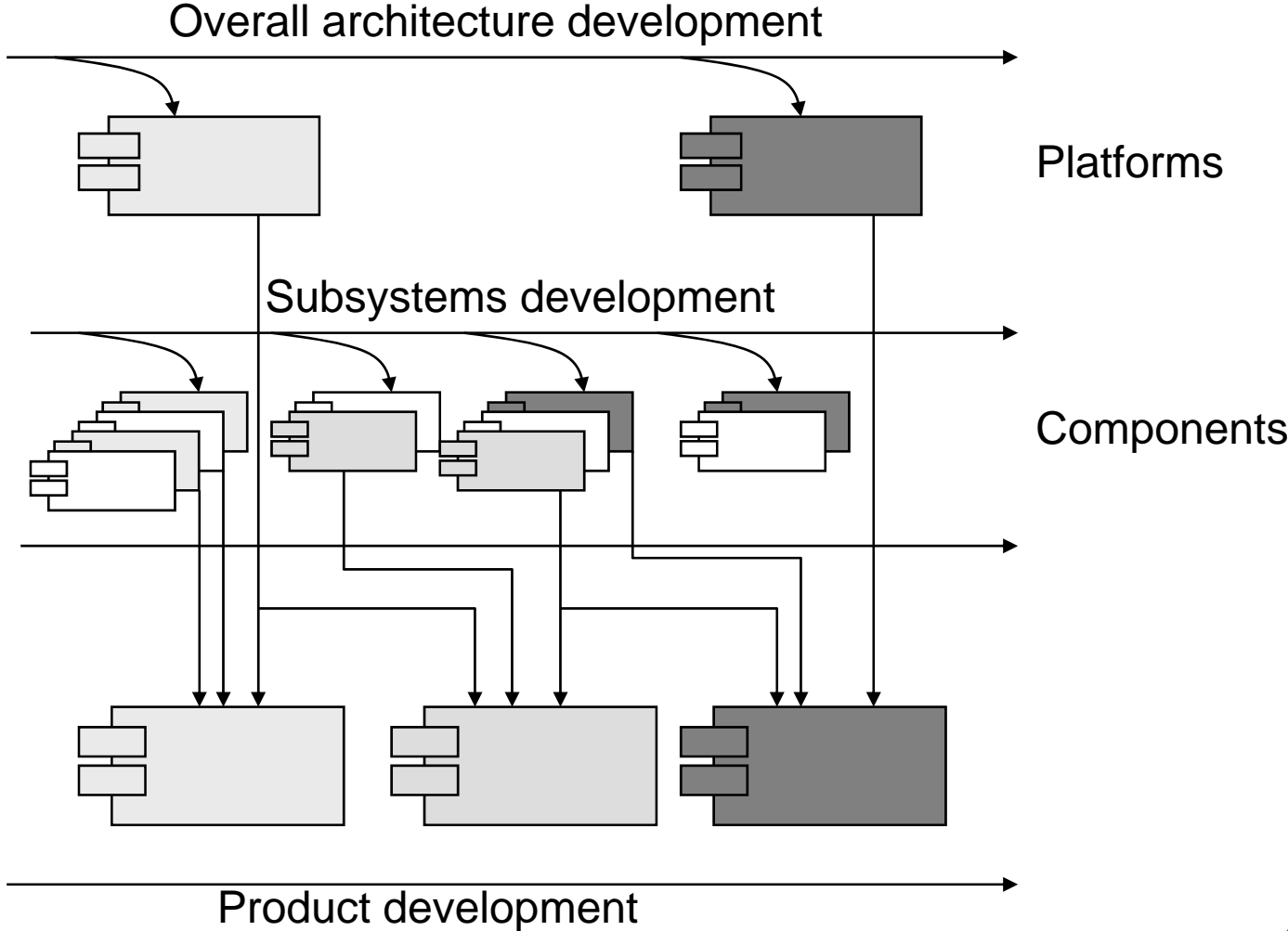
Product architecture:



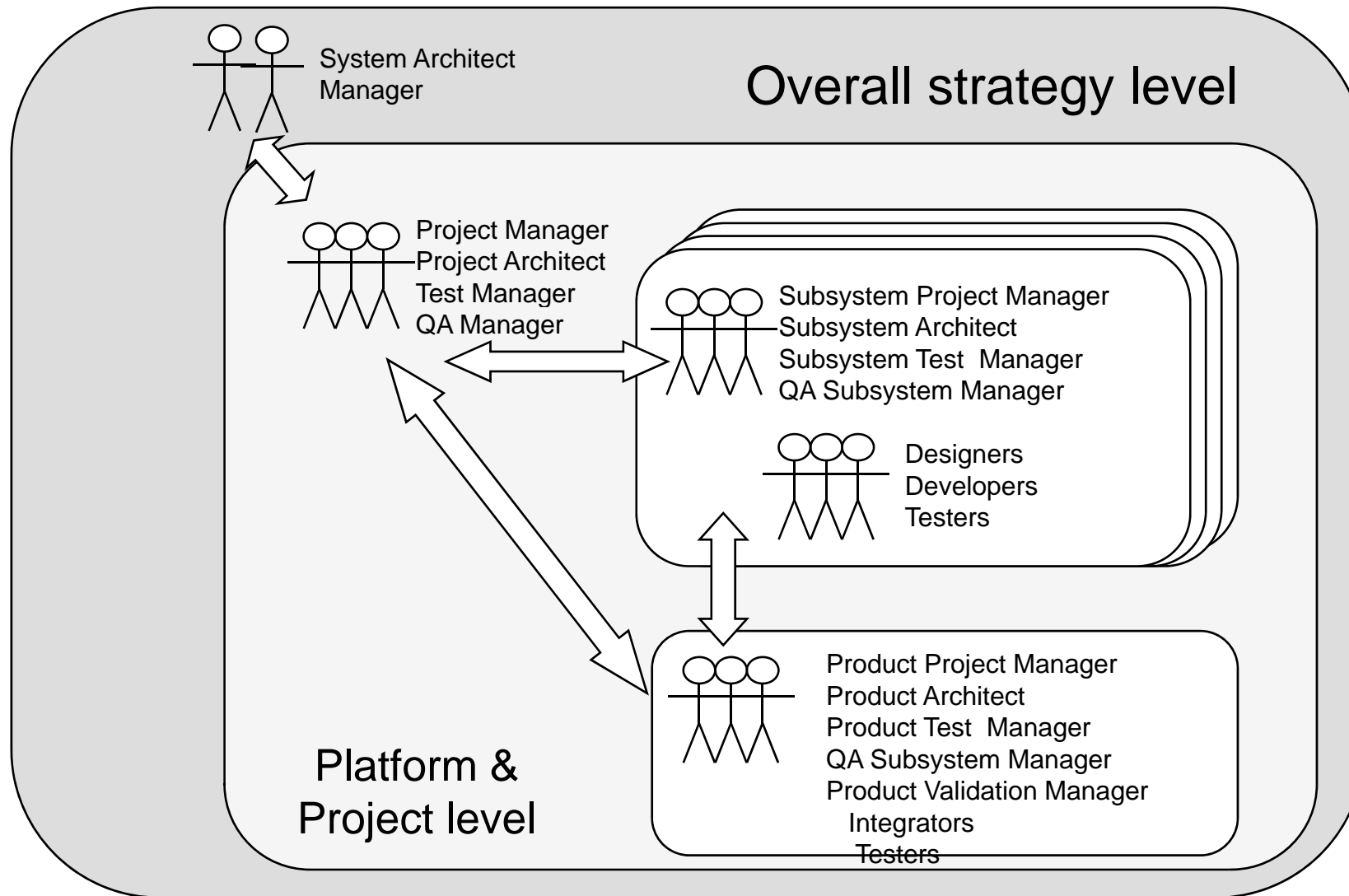
Koala Components



Development development process



CB development requires changes in the organizations!



Experience from Philips

- The new (CBD) approach did not work with the previous development process model and organization
- A lot of efforts has been put on
 - re-organization
 - Emphasis on early system/components specification
 - Quality assurance
 - Early test and verification of the components
 - Synchronization of the activities

Findings from the case study

- CBD requires specific approach in development process
- Reuse is not only a matter of a good technology but also of the process and organisation
- Many companies introducing CBD are not aware of that

Conclusion

- The main characteristic of component-base development process
 - a separation (and parallelization) of system development from component development.
- Consequence
 - Programming issues (low-level design, coding) are less emphasized
 - verification processes and infrastructural management requires significantly more efforts.

Literature

- Ivica Crnkovic, **Component-based Development Process and Component Lifecycle** (a chapter in a future CBSE book)
- *Component-based Development Process and Component Lifecycle*, Ivica Crnkovic, Stig Larsson, Michel Chaudron (*Technical University Eindhoven*), **CIT**,
<http://www.mrtc.mdh.se/index.phtml?choice=publications&id=0953>

ASSIGMENTS

- Write a report (a seminar work) / 15-20 pages.
- Alternatives
 - Component-based approach and agile methods
 - Component/based approach and RUP

 - Components and agent-based development (differences and similarities between agents and components)
 - Your own proposal related to your research