Unit-I

Process Models: Specialized

Specialized Process Models

Specialized process models

-Component-Based Development

-The Formal Methods Model

-Aspect-Oriented Software Development

- Team based Process Model

- Personal based Process Model

Specialized Process Models

COMPONENT BASED PROCESS MODEL

Commercial off-the-shelf (COTS) software components
– provide targeted functionality with well-defined interfaces
– components to be integrated into the software
•It incorporates many characteristics of the spiral model
•Evolutionary in nature
•an iterative approach to the creation of software
•Constructs applications from prepackaged software components
•Modeling and construction activities begin with the identification of candidate components
-suitable to OOD concept software

COMPONENT BASED PROCESS MODEL

-OOD software

This model incorporates the following steps:

1. Available component-based products are researched and evaluated for the application domain in question.
2. Component integration issues are considered.
3. A software architecture is designed to accommodate the components.
4. Components are integrated into the architecture.
5. Comprehensive testing is conducted to ensure proper functionality. - Software reuse, and reusability provides software engineers with a number of measurable benefits

The Formal Methods Model

- The formal methods model encompasses a set of activities that leads to formal mathematical specification of computer software
- It enables us to specify, develop, and verify a computer-based system by applying a rigorous, mathematical notation
- The variation in this approach – clean room software engineering
- Ambiguity, incompleteness, and inconsistency – can be corrected and discovered easily
- It is not a mainstream approach- but will give a defect free software

Specialized Process Models

The Formal Methods Model

Some pitfalls or disadvantages

- The development of formal models is currently quite time consuming and expensive.
- Because few software developers have the necessary background to apply formal methods, extensive training is required.
- It is difficult to use the models as a communication mechanism for technically unsophisticated customers.

Example:

*developers of aircraft avionics and medical devices

Specialized Process Models

Aspect-Oriented Software Development (aosd.net)

- Builders of complex software invariably implement a set of localized features, functions, and information content
- Localized software characteristics – modeled as components and constructed within the context of system architecture
- Modern computer-based systems are sophisticated – concerns —customer required properties or areas of technical interest—span the entire architecture
- Some Concerns – higher-level properties of the system

(e.g., security, fault tolerance)

Specialized Process Models

Aspect-Oriented Software Development

- When concerns cut across multiple system functions, features, and information - crosscutting concerns
- Aspectual requirements define those crosscutting concerns that have an impact across the software architecture.
- Aspect-oriented software development or Aspect oriented programming – it's a software engineering paradigm to define-specify-design-construct aspects
- "mechanisms beyond subroutines and inheritance for localizing the expression of a crosscutting concern"

Specialized Process Models

Aspect-Oriented Component Engineering (AOCE)

- AOCE uses a concept of horizontal slices through vertically-decomposed software components, called “aspects”
- to characterize cross-cutting functional and non-functional properties of components •Common Systemic aspects:
- User interfaces – viewing mechanism, extensible affordance and interface kind
- Collaborative work •Distribution - event generation, transport and receiving
- Persistency - data store/retrieve and indexing •Memory management
- Transaction processing - transaction atomicity, concurrency control and logging strategy
- Security - authentication, encoding and access right

Aspect-Oriented Component Engineering (AOCE)

- Evolutionary and concurrent model can be adopted
- Evolutionary – aspects are identified and constructed;
- Concurrent (parallel) – aspects are engineered independently of localized software components and yet, aspects have a direct impact on these components
- Instantiate asynchronous communication between the software process activities applied to the engineering and construction of aspects and components

THANK YOU