

# SWE 212: Software Engineering Processes

By

E. K. Olatunji (PhD)

Computer Sc Programme

FCAS

Thomas Adewumi University, Oke

April 2024

# Lecture Outline

- Course Contents
- Reference Materials
- Course Outlines
- First Pre-Lecture Assignment
- Preambles - Important Terms?
- Intro to Software Process (Activities)

# Course Contents

- This is introduction to Software Engineering Process
- – to be updated as we proceed
- **Course Contents**
- Software process definition- software process management and infrastructure, Software life cycles- categories of software processes, software life cycle models, software process adaption, practical considerations; Software process assessment and improvement – software process assessment methods, software process improvement models, and continuous and staged software process rating; Software measurement – software process and product measurement, quality of measurement results, and software process measurement techniques; Software engineering process tools.

# Course Objectives

- 1. To explain some important terms related to the course title
- 2. To discuss quality attributes of a good software
- 3. To describe activities involved in the software process
- 4. To describe the format and purposes of requirement specification document
- 5. To explain 4 generic process models with their adv, disadv & areas of applicability
- 6. To describe various forms of software measurement and metrics

# Learning Outcomes

- At the end of the course, students should be able to:
  - 1. explain some key terms related to the course title
  - 2. list & explain 4 attributes of a good software
  - 3. list & describe 4 important activities in SWE Process
  - 4. Describe purposes of requirement specification document
  - 5. List and explain 4 key generic process models
  - 6

# Important Reference Materials

- 1). Issues in software Engineering by Sodiya, A.S and Solanke O.O @ 2006, Abeokuta-Nigeria, Yomight Ventures
- 2). Software Engineering by I. Sommerville, 8ed, @ Pearson Education Limited, England. 2007, 2011
- 3). The requirements Engineering Handbook by Ralph R. Young @ Artech House Inc, Boston, UK, 2004
- Online Resources

# Course Contents Sequencing

- 

	SWE 212	Software Engineering Processes	
1	Week 1	Preambles- Important Terms	3 hours
2.	Week 2	Intro to Software Engineering Processes	3 hours
3.	Week 3-4	Software Process Activities	6 hours
4.	Week 5	Software Requirement Specification Documents	3 hours
5.	Week 6	CA1	3hours
6.	Week 7-8	Software Process models	6 hours
7.	Week 9	Software Measurement and Metrics	3 weeks
8		CA2	
9.			

# First Pre-Lecture Assignment

Q1) Explain the following terms:

(i) Software (ii) Engineering (iii) Software Engineering (SWE) (iv) scalability

• Q2) Write brief notes on the following terms or phrases:

- (software process (ii) CASE (iii) Software process models (iv) SWE methods

Q3) clearly distinguish between computer sc and SWE

Q4) List and explain 4 key attributes of a good software

- **Due date:** 1 working days to next lecture.
- **NB:** Always give the sources of your information. Citations and refs listing required in APA format



- Deliberately Left Blank

# SWE 212: Software Engineering Processes

By

E. K. Olatunji (PhD)

Computer Sc Programme

FCAS

Thomas Adewumi University, Oko

April 2024

# Lecture Outline

- Course Outline
- Reference Materials
- First Pre-Lecture Assignment
- Preambles - Important Terms?
- Intro to Software Process (Activities)

# Course Contents

- This is introduction to Software Engineering Process
- – to be updated as we proceed
- **Course Contents**
- Software process definition- software process management and infrastructure, Software life cycles- categories of software processes, software life cycle models, software process adaption, practical considerations; Software process assessment and improvement – software process assessment methods, software process improvement models, and continuous and staged software process rating; Software measurement – software process and product measurement, quality of measurement results, and software process measurement techniques; Software engineering process tools.

# Important Reference Materials

- 1). Issues in software Engineering by Sodiya, A.S and Solanke O.O @ 2006, Abeokuta-Nigeria, Yomight Ventures
- 2). Software Engineering by I. Sommerville, 8ed, @ Pearson Education Limited, England. 2007, 2011
- 3). The requirements Engineering Handbook by Ralph R. Young @ Artech House Inc, Boston, UK, 2004
- Online Resources

# First Pre-Lecture Assignment

Q1) Explain the following terms:

(i) Software (ii) Engineering (iii) Software Engineering (SWE) (iv) scalability

• Q2) Write brief notes on the following terms or phrases:

- (software process (ii) CASE (iii) Software process models (iv) SWE methods

Q3) clearly distinguish between computer sc and SWE

Q4) List and explain 4 key attributes of a good software

- **Due date:** 1 working days to next lecture.
- **NB:** Always give the sources of your information. Citations and refs listing required in APA format

- Deliberately Left Blank

# Preambles – Important Terms?

- What is Software
  - It is a collection of programs with its associated documentation and configuration data that are needed to make the program operate correctly
  - Configuration files are used to set up the programs; system documentation describes the structure of the system, while user-documentation explains how to use the system
- What is Engineering?
  - Engineering can be described as creative application of science, mathematical methods and empirical evidence to solve problems, such as design , construction and maintenance of structures, machines, materials, etc.



# Preambles – Important Terms?

- What is software Engineering(SWE)?
  - SWE is the application of engineering techniques and principles to the development/production of high quality software system/software product.
  - Software development are considered as engineering discipline becos it involves most, if not all, the activities that exist in the physical engineering discipline
  - In the physical engineering discipline, the following activities exist: Create (?), design, construction, implementation, supervision, assessment, rehabilitation, extension – (Sodiya & Solanke). All these and much more also happen in software development

# Preambles – Important Terms Contd-1?

- Difference between Software engineering and Computer Science
  - CSC is concerned with the theories & methods that underlie computer and software systems
  - SWE is concerned with the practicalities of developing and delivering of useful software
- Look for more!! i.e elaborate with examples

# Preambles – Important Terms Contd-1?

- What is Software Process?
  - It is a set of activities whose goal is the development or evolution of software product
  - There are 4 fundamental process activities common to all software processes:
    - Software specification – specification of the duties/services of the software and the constraints on its operation
    - Software development – design and implementation of the system
    - Software validation – testing to ensure the system does what the user wants
    - Software evolution – modification of the software to make it adapt to changing customer and market requirements/demands

# Preambles – Important Terms Contd-2?

- What is software process Model?
  - This is a simplified description of a software process that presents one view of that process – actually an approach to software development
  - Most software process models are based on one of the 3 generic methods or paradigms of software development:
    - The **waterfall** model/approach – the main 4 process activities are done sequentially, one after the other
    - **Iterative development** model – the process activities are carried out in parallel
    - Component-based software engineering(**CBSE**) model – existing system components are integrated together rather than developing a system from scratch

# Preambles – Important Terms Contd-2?

- What is CASE (Computer Assisted software engineering)?
  - CASE stands for computer-Aided Software Engineering.
  - It cover a wide range of different types of programs that are used to support software process activities such as requirement analysis, system modelling, debugging and testing
- Two categories of Software product
  - Generic product – e.g., word processor, drawing packages, project management tools
    - They are stand alone systems that are produced by a development organization/software house and sold on the open market
    - It is the developing organization that controls the specification of this type of software
  - Custom /customized Product – these are systems commissioned by a particular org
    - E,g; a system written to support a particular business process. Air traffic control system
    - Its specification is controlled by the organization requesting for the software

# Preambles – Important Terms Contd 3?

- What are the attributes of a good software?
  - Maintainability – Ease of making changes to the software
  - Dependability – (including reliability, security, safety)
    - A dependable software should not cause physical or economic damage in the event of system failure
  - Efficiency – This includes responsiveness, processing time, memory usage, etc
  - Usability – must be usable without undue effort, by the users for which it is developed – It should have an appropriate user interface and adequate documentation

# Preambles – Important Terms Contd 3?

- What are software engineering methods?
  - A SWE method is a structured approach to software development whose aim is to facilitate the production of high-quality software in a cost effective way.
  - Methods include:
    - Structured Analysis
    - Object-oriented method
    - Unified Modelling language (UML)

# Second Pre-Lecture Assignment

- Write on detailed activities involved in :
  - (i) Software process
  - (i) Requirement Engineering Process
  - (ii) Software Design
- **Due date:** 1 working days to next lecture.
- **NB:** Always give the sources of your information. Citations and refs listing required in APA format



- Deliberately Left Blank

# Third Pre-Lecture Assignment

# Four Pre-Lecture Assignment

# SWE 212: Software Engineering Processes

By

E. K. Olatunji (PhD)

Computer Sc Programme

FCAS

Thomas Adewumi University, Oke

April 2024

# Lecture Outline

- Course Outline
- Reference Materials
- First Pre-Lecture Assignment
- Preambles - Important Terms?
- Intro to Software Process (Activities)

# Course Contents

- This is introduction to Software Engineering Process
- – to be updated as we proceed
- **Course Contents**
- Software process definition- software process management and infrastructure, Software life cycles- categories of software processes, software life cycle models, software process adaption, practical considerations; Software process assessment and improvement – software process assessment methods, software process improvement models, and continuous and staged software process rating; Software measurement – software process and product measurement, quality of measurement results, and software process measurement techniques; Software engineering process tools.

# Important Reference Materials

- 1). Issues in software Engineering by Sodiya, A.S and Solanke O.O @ 2006, Abeokuta-Nigeria, Yomight Ventures
- 2). Software Engineering by I. Sommerville, 8ed, @ Pearson Education Limited, England. 2007, 2011
- 3). The requirements Engineering Handbook by Ralph R. Young @ Artech House Inc, Boston, UK, 2004
- Online Resources

# Previous Lecture Notes

- Important Terms



# Second Pre-Lecture Assignment

- Write on detailed activities involved in :
  - (i) Software process
  - (i) Requirement Engineering Process
  - (ii) Software Design
- **Due date:** 1 working days to next lecture.
- **NB:** Always give the sources of your information. Citations and refs listing required in APA format

- Deliberately Left Blank

# Intro to Software Process

- There are 4 main activities involved in the software process : software specification, software design and program development, software validation and software evolution.

## □ Software Specification

- This is also called Requirements Engineering or System Requirements definition
- System requirement definition specify what the system should do (its functions and services) and its essential and desirable system properties
- In creating system requirements definitions, you will need to consult with system customers and end-users
- Three Types of requirements:
  - Abstract functional requirement
  - System properties – i.e non-functional emergent system properties such as availability, performance and safety
  - Features that a system must not exhibit; e,g an air traffic control system might say that the system should not present the controller with too much information
- Non-functional/Emergent properties of a system
  - This includes reliability, performance, safety, security, usability, reliability (hardware, software, operator)
  - Operator reliability – i.e how likely is it that the operator of a system will make error

# Intro to Software Process Contd-1

## □ Software Design and program Development (i.e Design and Implementation)

- This stage of software development consists of software design and programming
- A software design is a description of :
  - (1) the structure of the software to be implemented (2) the data that is part of the system (3) the interfaces between the system components (4) sometimes the algorithm used
- Designers do not arrive at a finished product all at once but develop the design iteratively thru a no. of versions/revisions

# Intro to Software Process Contd-2

## ❑ Software Validation (system testing and integration)

- Software validation is intended to ensure that a system conforms to its specification and that the system meets the expectations of the users
- It involves testing processes at each stage of the software process from user requirement definition to program development

## ❑ Software Evolution (Software Maintenance)

- This has to do with making changes to the developed system in order to make the system still relevant in satisfying the users' constant changing needs / requirements

- Deliberately left blank

# Software Process Activities

- There are 4 main activities involved in the software process : software specification, software design and program development, software validation and software evolution.

## □ Software Specification

- This is also called Requirement Engineering / software Requirement specification
- It is the process of understanding and defining what services are required from the system and identifying the constraints on the system operations and deployment
  - (i.e. the functional and non functional requirements)
- There are 4 main phases in the requirement engineering process:
  - Feasibility study, Requirement elicitation and analysis, Requirement specification and Requirement validation

# Software Process Activities Contd-1

- There are 4 main phases in the requirement engineering process:
  - Feasibility study, Requirement elicitation and analysis, Requirement specification and Requirement validation
- **Feasibility study**
  - This study is carried out to ascertain whether the identified needs of the user can be satisfied using existing software and hardware technology and within the existing budgetary constraints
- **Requirement Elicitation and Analysis**
  - This is the process of determining the system requirements thru observation of existing system, discussion with potential users, task analysis, etc



# Software Process Activities Contd-2

- There are 4 main phases in the requirement engineering process:
  - Feasibility study, Requirement elicitation and analysis, Requirement specification and Requirement validation
- **Requirement specification**
  - The info gathered during analysis activity is translated into a document that defines a set of requirements
- **Requirement validation**
  - At this stage, documented system requirements are checked for realism, consistency and completeness
  - Any discovered error or omission is also corrected
- **NB:** The activities of analysis, definition and specification are interleaved

# Software Process Activities Contd-3

## □ Software Design and program Development (i.e Design and Implementation)

- This stage of software development consists of software design and programming
- A software design is a description of :
  - (1) d structure of the software to be implemented (2) d data that is part of the system (3) the interfaces between the system components (4) sometimes the d algorithm used
- Designers do not arrive at a finished product all at once but develop the design iteratively thru a no. of versions/revisions
- The specific design process activities are:
  - Architectural design –
  - Abstract specification –
  - Interface design –
  - Component design –
  - Data structure design –
  - Algorithm design -

# Software Process Activities Contd-4

- The specific design process activities are:
  - **Architectural design** – the sub systems making up the system and their relationships are identified and documented
  - **Abstract specification** – for each sub system, an abstract specification of its services and constraints under which it must operate is produced
  - **Interface design** – (including I/O interfaces). For each sub system, its interfaces with other subsystems is designed and documented
  - **Component design** (program component?) – services are allocated to components and the interfaces of these components are designed
  - **Data structure design** (file/database design) – The data structure that will be used in the system implementation are designed in details and specified
  - **Algorithm design** (program Spec?) – The algorithm used to provide services are designed in detail and specified

# Software Process Activities Contd-4b

- More on Software Design Process Activities
- **3. System Design**
- A blue print of the desired Info System is provided
- The following elements are candidates for design
  - **User-interfaces-**
    - ✓ bcos users may need to interact with the system;
    - ✓ various screens have to be designed together with associated processing operations
  - **Outputs;**
    - ✓ GUIs are designed;
    - ✓ Reports design(forms/formats, types, vol, frequency of reports);
    - ✓ choice of output media
  - **Inputs –**
    - ✓ types of input media;
    - ✓ design of input forms and layouts
  - **Files & database components –**
    - ✓ Consideration for storage media;
    - ✓ method of file/database org &access; layout of records/tables
  - **Procedures & programs –** all programs needed to produce the desired output from the given input

•

# Software Process Activities Contd-5

## ❑ Software Validation (system testing and integration)

- Software validation is intended to ensure that a system conforms to its specification and that the system meets the expectations of the users
- It involves testing processes at each stage of the software process from user requirement definition to program development
- The 3 main stages of testing processes are :
  - **Component (or unit) testing**
    - Individual component is tested to ensure it operates correctly
    - A component may be a simple entity like functions, object classes, and a group of these entities
  - **System testing**
    - The components are integrated to makeup the system
    - The process is concerned with validating that d system meets its functional and non-functional requirements as well as testing the emergent system properties
  - **Acceptance testing**
    - This is the final stage in the testing process before the system is accepted for operational use
    - The system is tested with data supplied by the system customer rather than with simulated test data of the developer/programmer

# Software Process Activities Contd-6

- **Note:** component development and testing are normally interleaved
- Acceptance testing is sometimes called **alpha (a)** testing. Custom systems are developed for a single client.
  - The alpha testing process continues until the system developer and the client agree that the delivered system is an acceptable implementation of the system requirements
  - When a system is to be marketed as a software product, a testing process , called **beta (B)** testing is often used

## □ Software Evolution (Software Maintenance)

- This has to do with making changes to the developed system
- Software engineering (SWE) can realistically be thought of as an evolutionary process where software is continually modified over its lifetime in response to changing system requirements and customers' needs.

# Third Pre-Lecture Assignment (Revised)

- 1). Distinguish between Software process and software process model
- 2). Write exhaustively on, at least 5 software process models.

Distinguish among them

- 3). What do you understand by software requirement specification document?
- Dsg??
- **Due date:** 2 working days to next lecture.
- **NB:** Always give the sources of your information. Citations and refs listing required in APA format

- Deliberately Left Blank



# SWE 212: Software Engineering Processes

By

E. K. Olatunji (PhD)

Computer Sc Programme

FCAS

Thomas Adewumi University, Oke

April 2024

# Lecture Outline

- Course Outline
- Reference Materials
- First Pre-Lecture Assignment
- Preambles - Important Terms?
- Intro to Software Process (Activities)

# Course Contents

- This is introduction to Software Engineering Process
- – to be updated as we proceed
- **Course Contents**
- Software process definition- software process management and infrastructure, Software life cycles- categories of software processes, software life cycle models, software process adaption, practical considerations; Software process assessment and improvement – software process assessment methods, software process improvement models, and continuous and staged software process rating; Software measurement – software process and product measurement, quality of measurement results, and software process measurement techniques; Software engineering process tools.

# Important Reference Materials

- 1). Issues in software Engineering by Sodiya, A.S and Solanke O.O @ 2006, Abeokuta-Nigeria, Yomight Ventures
- 2). Software Engineering by I. Sommerville, 8ed, @ Pearson Education Limited, England. 2007, 2011
- 3). The requirements Engineering Handbook by Ralph R. Young @ Artech House Inc, Boston, UK, 2004
- Online Resources

# Previous Lecture Notes

- Important Terms
- Intro to Software Engineering Process
- **Software Process Activities** : Specification, Design & Implementation, Validation/Testing and Evolution/Maintenance

# Third Pre-Lecture Assignment ()

- 1). Write briefly on Software Design Activities
- 2) Distinguish between Software process and software process model
- 3). Write exhaustively on, at least 5 software process models.  
Distinguish among them
- 4). What do you understand by software requirement specification document?
- Dsg??
- **Due date:** 2 working days to next lecture.
- **NB:** Always give the sources of your information. Citations and refs listing required in APA format

- Deliberately Left Blank

# Software Requirements Specification Document

(from sommerville and Sodiya& co)

- The requirement engineering process leads to the production of a requirement document
- The requirements are usually presented at 2 levels of detail/granularity:
  - End-user & customer need a high level statement of the requirement
  - Developers need a more detailed requirement specification
- The software requirements specification (**SRS**) document states in clear and precise language the **functions & capabilities** a software system must provide; as well as any required **constraints** by which the system must abide



# Software Requirements Specification (SRS) Document Contd-1

## □ Purpose of SRS document

- 1. it provides a feedback to the customer, assuring them that the developer understands the issues or problem to be solved
- 2. it decomposes the problem into its component parts
- It serves as input to the design specification
- 4. it serves as a product validation check

# Software Requirements Specification (SRS) Document Contd-2

## □ SRS Techniques

- The requirement specialists do their work by (1) talking to people, (2) documenting their findings, (3) analysing the collected info to discover inconsistencies or oversights; and then (4) talk to people again
- Info gathering techniques include:
  - Onsite visitation ( to observe users at work), also called **ethnography**?!
  - Document inspection
  - Questionnaires
  - Surveys
  - Stake holder's interview
  - Requirement workshop / seminar
    - Also called Joint Requirement Development(JRD) session, where requirement are jointly identified and defined
  - Etc.

# Software Requirements Specification (SRS) Document Contd-3

## □ IEEE Template for writing SRS Documents from Solanke & Sodiya

- Section 1. Introduction
- Section 2. Overall Description
- Section 3. External Interface requirements
- Section 4. System Features
- Section 5. Other Non-functional Requirements
- Section 6. Other Requirements
- Appendix A: Terminologies/Glossary/ Definition of Terms
- Appendix ?? : To be determined

# Software Requirements Specification (SRS) Document Contd-4

## □ IEEE Template for writing SRS Documents from Solanke & Sodiya

- Section 1. Introduction
  - Purpose, Document conventions, Intended Audience, Additional Info, SRS team members, References
- Section 2. Overall Description
  - Product perspectives, Product Functions, User classes and characteristics, Operating Environment, User environment, Design Implementation constraints, Assumptions and dependencies
- Section 3. External Interface requirements
  - User interfaces, Hardware Interfaces, Software Interfaces, Communications Protocols and interfaces,
- Section 4. System Features
- Section 5. Other Non-functional Requirements
- Section 6. Other Requirements
- Appendix A: Terminologies/Glossary/ Definition of Terms
- Appendix ??: To be determined

# Software Requirements Specification (SRS) Document Contd-4

## □ IEEE Template for writing SRS Documents from Solanke & Sodiya

- Section 1. Introduction
- Section 2. Overall Description
- Section 3. External Interface requirements
- Section 4. System Features
  - System Feature A – Description and priority, Action Result, Functional Requirements
  - System Feature B-
- Section 5. Other Non-functional Requirements
  - Performance Requirements, Safety Requirements, Security Requirements , Software Quality attributes, Product Documentation, User Documentation,
- Section 6. Other Requirements
- Appendix A: Terminologies/Glossary/ Definition of Terms
- Appendix ?? : To be determined

# Four Pre-Lecture Assignment

- 1). Distinguish between Software process and software process model
- 2). Write exhaustively on, at least 5 software process models.  
Distinguish among them
- 3). ??
- **Due date:** 2 working days to next lecture.
- **NB:** Always give the sources of your information. Citations and refs listing required in APA format

- Deliberately left blank

# SWE 212: Software Engineering Processes

By

E. K. Olatunji (PhD)

Computer Sc Programme

FCAS

Thomas Adewumi University, Oke

April 2024



# Lecture Outline

- Course Outline
- Reference Materials
- First Pre-Lecture Assignment
- Preambles - Important Terms?
- Intro to Software Process (Activities)

# Course Contents

- This is introduction to Software Engineering Process
- – to be updated as we proceed
- **Course Contents**
- Software process definition- software process management and infrastructure, Software life cycles- categories of software processes, software life cycle models, software process adaption, practical considerations; Software process assessment and improvement – software process assessment methods, software process improvement models, and continuous and staged software process rating; Software measurement – software process and product measurement, quality of measurement results, and software process measurement techniques; Software engineering process tools.

# Important Reference Materials

- 1). Issues in software Engineering by Sodiya, A.S and Solanke O.O @ 2006, Abeokuta-Nigeria, Yomight Ventures
- 2). Software Engineering by I. Sommerville, 8ed, @ Pearson Education Limited, England. 2007, 2011
- 3). The requirements Engineering Handbook by Ralph R. Young @ Artech House Inc, Boston, UK, 2004
- Online Resources

# Previous Lecture Notes

- Important Terms
- Intro to Software Engineering Process
- Software Process Activities
- Software requirement Specification Document

# Fourth Pre-Lecture Assignment (Revised)

- 1). Write brief notes on Software Design Activities
- 2). Distinguish between Software process and software process model
- 3). Write exhaustively on, at least 5 software process models.  
Distinguish among them
- 4).??
- **Due date:** 2 working days to next lecture.
- **NB:** Always give the sources of your information. Citations and refs listing required in APA format

- Deliberately Left Blank

# Software Process Models

- Contents
- a) Summary of software Process Model (SPM)
- b) Detail of the 3 basic models
  - What? steps,? Adv/Disadv?
- C) Variations of the Evolutionary Development (/Iterative development)
  - Throwaway prototyping
  - Exploratory /Incremental delivery
  - Spiral model
  - V-Model (Validation-based Model)
    - The what? Steps & Adv / Disadv? Of the 4 variations above
- hg

# Software Process Models Contd-1

- A software process model is a simplified description of a software process that presents one view of that process
- ❑ **Summary of the basic Generic model**
- There are 3 basic generic Software Process Model (or process paradigm)
- **a) The Water Fall Model**
  - In this model, the 4 basic process activities are taken as separate process phases and are carried out sequentially. It goes from requirements specification, design, implementation, testing, deployment and maintenance in this order,
  - Each phase of this model is usually completed before the next phase begins
- **b) Evolutionary development (or Iterative development) Model**
  - In this model/approach, the process activities of specification, design, implementation, integration & testing are carried out in parallel within a development activity.
  - An initial system is rapidly developed from a very abstract specification. This is then refined with customer input to produce a system that satisfies the customers needs
- **c) Component-Based software Engineering (or Re-use Model SWE)**
- **d) Cleanroom Approach/Model**
- e) ??



# Software Process Models Contd-2

- A software process model is a simplified description of a software process that presents one view of that process
- Summary of the basic Generic model Contd
- There are 3 basic generic Software Process Model (or process paradigm)
- a) The Water Fall Model
- b) Evolutionary development (or Iterative development) Model
- c) Component-Based software Engineering (or Re-use Model SWE)
  - In this approach, it is assumed that some parts of the system already exists. The system development process focuses on integrating the parts/components rather than developing the system from scratch.
- d) Cleanroom Approach/Model
  - A variant of the 3 generic process models described above is the cleanroom model/approach
  - In d approach, a formal model is created which is then transformed , using some mathematical formulae that preserves its consistency, into executable code.
  - In d cleanroom process, each software increment is formally specified and this specification is transformed into an implementation.
  - Software correctness is demonstrated using a formal approach
- e) ??

# Software Process Models Contd-3

## ❑ Water Fall Model

- What?
  - The key activities in this model are Requirement analysis specification, System Design, Implementation (Program development)& Unit testing, Integrating & System testing, deployment and maintenance
  - Each of these activities is carried out sequentially, one after the other
- Adv?
  - Documentation is produced at each stage; thus no room for confusion
  - System requirements are identified long before programming begins
  - Development team can easily adapt to the practices
- Disadv?
  - Its inflexible partitioning of the project does not allow it to respond to changing user requirements
  - Product delivery takes a long time

# Software Process Models Contd-4

## ❑ Evolutionary Development/Iterative development (Cf Agile method in SA&D??)

- What?
  - This model is based on the idea of developing an initial implementation, exposing this to user/customer & refining it thru many versions until an adequate/satisfactory system has been developed.
  - Specification, development& validation activities are carried out in parallel rather than being separate phase, with rapid feedback across activities.
  - This approach to software development is often more effective than waterfall approach in producing a system that meets immediate needs of customers
- Step?
- Adv?
  - The specification can be developed incrementally
  - Faster delivery time for the software project
  - Faster and adaptable to changes much more rapidly
- Disadv? From an engineering pt of view
  - The process is not visible; managers need regular deliberations to monitor progress
  - System are often poorly structured: continual change tends to corrupt the software structured, - maintenance becomes increasingly difficult & costly.
  - Documentation gets left behind

# Software Process Models Contd-5

## ❑ Component-Based SWE

- What?

- This model relies heavily on re-use of existing software code. The process consists of the following activities:

- **Component Analysis**

- ✓ Given a requirement specification, a search is made for components to implement that specification

- **Requirement modification**

- ✓ At this stage, requirements are analyzed using info about the component that have been discovered, They are then modified to reflect the available component

- **System design with re-use**

- **Development & integration**

- Step?
- Adv?
- Disadv?

# Software Process Models Contd-5

## ❑ Component-Based SWE

- What?
  - This model relies heavily on re-use of existing software code. The process consists of the following activities:
    - Component Analysis
    - Requirement modification
    - System design with re-use
      - ✓ During this phase, a framework of the system is designed or an existing framework is reviewed. The designers take into account the components that are re-used & organize the framework to cater for this
      - ✓ Some new software may have to be designed, if re-usable components are not available.
    - Development & integration
      - ✓ System that cannot be externally procured is developed, and the components & COTS (Commercial off-the-shelf) systems are integrated to create the new system.
- Step?
- Adv?
- Disadv?

# Software Process Models Contd-6

## □ Component-Based SWE Continued

- What?
- Step?
- Adv?
  - Reduces amount of software to be developed – thus reducing cost & risk
  - Usually leads to faster delivery of the software
  - Adaptable and flexible to changes
- Disadv?
  - Some requirements may have to be compromised – thus users may not be 100% / completely satisfied.

# Fifth Pre-Lecture Assignment

- 1) Distinguish between Software process and software process model
- 2) Discuss on at least 5 variants of the iterative model; high-lighting the what ? The steps? Advantages ?and disadvantages ? of each
  
- **Due date:** 2 working days to next lecture.
- **NB:** Always give the sources of your information. Citations and refs listing required in APA format

- Deliberately left blank



# SWE 212: Software Engineering Processes

By

E. K. Olatunji (PhD)

Computer Sc Programme

FCAS

Thomas Adewumi University, Oke

April 2024

# Lecture Outline

- Course Outline
- Reference Materials
- First Pre-Lecture Assignment
- Preambles - Important Terms?
- Intro to Software Process (Activities)

# Course Contents

- This is introduction to Software Engineering Process
- – to be updated as we proceed
- **Course Contents**
- Software process definition- software process management and infrastructure, Software life cycles- categories of software processes, software life cycle models, software process adaption, practical considerations; Software process assessment and improvement – software process assessment methods, software process improvement models, and continuous and staged software process rating; Software measurement – software process and product measurement, quality of measurement results, and software process measurement techniques; Software engineering process tools.

# Important Reference Materials

- 1). Issues in software Engineering by Sodiya, A.S and Solanke O.O @ 2006, Abeokuta-Nigeria, Yomight Ventures
- 2). Software Engineering by I. Sommerville, 8ed, @ Pearson Education Limited, England. 2007, 2011
- 3). The requirements Engineering Handbook by Ralph R. Young @ Artech House Inc, Boston, UK, 2004
- Online Resources

# Previous Lecture Notes

- Important Terms
- Intro to Software Engineering Process
- Software Process Activities
- Software requirement Specification Document
- Software Process Models

# Fifth Pre-Lecture Assignment

- 1) Distinguish between Software process and software process model
- 2) Discuss on at least 5 variants of the iterative model; high-lighting the what ? The steps? Advantages ?and disadvantages ? of each
  
- **Due date:** 2 working days to next lecture.
- **NB:** Always give the sources of your information. Citations and refs listing required in APA format

- Deliberately Left Blank

# Software Process Models Contd-7

## □ Variations of Iterative (/ Evolutionary) Development

- There are a number of variations of the evolutionary/iterative development ? These include:
- Throwaway prototyping
- Exploratory / Incremental delivery
- Spiral model
- V-model



# Software Process Models Contd-8

## ❑ Throwaway prototyping (from Solanke&Co)

### • What?

- This model was developed on the assumption that it is often difficult to know all the requirements at the beginning of a project
- The model offers a development approach that yields results without 1<sup>st</sup> requiring all info up-front
- In this model, the developer builds a simplified version of the proposed system and presents it to the customer for consideration. The customer in turn provides feedback to the developer who goes back to refine the system requirements to incorporate the additional info.
- Often the prototype code is thrown away and entirely new programs are developed once requirements are fully identified

### • Steps

- Requirement definition/collection – the info collected is usually limited to a subset of the complete system requirements
- Design (based on the requirements known)
- Prototype creation/modification
- Assessment. The prototype is presented to the customer for review. Comments & suggestions are collected from them
- Prototype refinement. The earlier prototype is revised to make it more effective
- System implementation. The system is re-written once requirements are understood.

\* The iteration process eventually produces a working system!

### • Adv? – mine!

### • Disadv?

# Software Process Models Contd-9

## ❑ Throwaway prototyping (from Solanke&Co)

### • What?

- In this model, the developer builds a simplified version of the proposed system and presents it to the customer for consideration. The customer in turn provides feedback to the developer who goes back to refine the system requirements to incorporate the additional info.
- Often the prototype code is thrown away and entirely new programs are developed once requirements are fully identified

### • Steps?

### • Adv? – mine!

- Quick delivery of a subset of the product
- Flexibility – since customer feedback can easily be incorporated

### • Disadv

- Prototyping can lead to false expectations in that the customer can think development is complete with a prototype
- It can lead to a poorly designed system
- Another disadvantage is maintenance difficulty

### • Variation of prototyping

- Rapid Application Development (RAD) is a variation of prototyping in which strict time limits are imposed on each development phase; and it relies heavily on rapid application tools which enable quick development

# Software Process Models Contd-10

## □ Incremental delivery / (Exploratory?)

### • What?

- In this model, services of d system are identified & high-lighted in outline form
- Priority is attached to the different services
- A no of delivery increment are defined with each increment providing a subset of entire system functionality
- The service with d highest priority are delivered first

### • Steps

### • Adv?

- Customers do not have to wait until d entire system is delivered before gaining value from it
- There is lower risk of overall project failure
- Customers are less likely to encounter software failure in the most important parts of the system. This is bcos d services with the highest priority are delivered first having been tested thoroughly.

### • Disadv

- Mapping of customers' requirements to increment of the right size can be very difficult
- It can be difficult to identify common functions/facilities that are required by all increments since requirements are not defined at once

### • Variation

- A variant of the incremental approach is called **extreme programming (XP)**

### • etc

# Software Process Models Contd-11

- **Spiral model**
  - The main difference between the spiral model & other models is the explicit recognition of risk in the spiral model
  - Informally, risk means something that can go wrong, For eg, if the intention is to use a new programming lang (PL), a risk may be that the available compilers are unable or do not produce sufficiently efficient object code
- **What?**
  - The software process in spiral model development is represented as a spiral or concentric circles or loop
  - Each loop in the spiral represents a phase of the software process
  - Thus the innermost loop might be concerned with system feasibility, the next loop with requirement definition, the next loop with system design, etc.
  - Each loop in the spiral is split into 4 sectors
  - See fig 4.5 of Sormerville text book!
- **Steps**
  - Objective setting
  - Risk assessment & reduction
  - Development and validation
  - Planning & management
- **Adv?**
- **Disadv**

# Software Process Models Contd-12

- Spiral model Contd
- What?
- Steps
  - Objective setting
    - Objs are determined, possible obstacles identified, and alternative approaches are weighed
  - Risk assessment & reduction
    - For each identified project risks, detail analysis is carried out
    - Steps to reduce the risk is taken
    - Sometimes prototyping is used to clarify needs
  - Development and validation
  - Planning & management
- Adv?
  - Assignment for you
- Disadv
  - Assignment for you
- etc

# Software Process Models Contd-12

- Spiral model Contd
- What?
- Steps
  - Objective setting
  - Risk assessment & reduction
  - Development and validation
    - After risk evaluation and reduction, a development model for the system is chosen depending on the nature of the dominant risk
    - For e.g; if user-interface risks are dominant; an appropriate development model might be evolutionary prototyping
    - If safety risks are the main consideration, development based on formal transformation may be the most appropriate, etc
  - Planning & management
    - The project is reviewed & decision made whether to continue with a further loop of the spiral.
    - If the decision is to continue, plans are drawn up for the next phase of the project
- Adv?
  - Assignment for you
- Disadv
  - Assignment for you
- etc

# Software Process Models Contd-13

- **V—Model (Validation Model)**
  - This is an elaboration of the waterfall model and stresses the necessity of validation activities
- What?
  - Each step of the waterfall model has a matching validation process which can, where defects are found, cause a loop back to the corresponding development stage and a reworking of the following step
- Steps
- Adv?
  - Assignment for you
- Disadv
  - Assignment for you
- etc

# Six Pre-Lecture Assignment

- By going to appropriate units of the (TAU) University, you are to give:
    - (i) Software Requirements specification and
    - (ii) a fair system design
    - of the following systems:
  - A) Hostel Allocation system (System of allocating hostel to students)
  - B) TAICO secondary school management system
  - C) TAU Library Management System
- 
- **Due date:** 3/4 weeks from the time assignment is given
  - **NB:** Always give the sources of your information. Citations and refs listing required in APA format





- Deliberately left blank

# Sample System Specification and Design

- Would be sent to you soonest!

# SWE 212: Software Engineering Processes

By

E. K. Olatunji (PhD)

Computer Sc Programme

FCAS

Thomas Adewumi University, Oke

April 2024

# Lecture Outline

- Course Contents
- Reference Materials
- Previous Topics Covered
- Sixth Pre-Lecture Assignment
- Software metrics and Measurement

# Course Contents

- This is introduction to Software Engineering Process
- – to be updated as we proceed
- **Course Contents**
- Software process definition- software process management and infrastructure, Software life cycles- categories of software processes, software life cycle models, software process adaption, practical considerations; Software process assessment and improvement – software process assessment methods, software process improvement models, and continuous and staged software process rating; Software measurement – software process and product measurement, quality of measurement results, and software process measurement techniques; Software engineering process tools.

# Important Reference Materials

- 1). Issues in software Engineering by Sodiya, A.S and Solanke O.O @ 2006, Abeokuta-Nigeria, Yomight Ventures
- 2). Software Engineering by I. Sommerville, 8ed, @ Pearson Education Limited, England. 2007, 2011
- 3). The requirements Engineering Handbook by Ralph R. Young @ Artech House Inc, Boston, UK, 2004
- Online Resources

# Previous Lecture Notes

- Important Terms
- Intro to Software Engineering Process
- Software Process Activities
- Software requirement Specification Document
- Software Process Models
- Variants of Iterative /Evolutionary Development models



# Sixth Pre-Lecture Assignment

- 1) Distinguish between the following pairs of Software metrics (i) process and Product metrics (ii) Control and Predictor metrics (iii) dynamic and Static metrics
  - 2) Describe 2 examples of each category of software metrics in (1) above
  - 3) why do we need to measure software processes and/ or products?
- 
- **Due date:** 2 working days to next lecture.
  - **NB:** Always give the sources of your information. Citations and refs listing required in APA format

- Deliberately Left Blank

# Software Measurement & Metrics

## □ References

- SWE by Ian Sommerville, 10ed, @ Pearson Education Limited, 2016, England
- Why Measure Software? (Extracted from:  
[https://www.cs.purdue.edu/homes/cs307/slides/lecture14\\_2up.pdf](https://www.cs.purdue.edu/homes/cs307/slides/lecture14_2up.pdf) on 19-06-2024
- Other online resources

# Software Measurement & Metrics

- **Introduction**

- **What is Measurement?**

- Measurement is the act of measuring, or quantifying or making a quantitative assessment of something
- Measurements are more objective since they may be compared and recorded against numerical standards
- The process of figuring out a physical object's characteristics and size is known as measurement.
- Observations that can be stated numerically—are accounted for via measurement.
- For example, measurement examines something's size in numerical terms, including how much, how tall, how quickly, how hot, how far away, and how little it is
- For instance, if you want to know someone's height, you can measure them with a tape measure and get a number like 188cm. This is a measurement of this person's height.

# Software Measurement & Metrics

- **Introduction Contd**

- Measurement is the act of measuring, or quantifying or making quantitative assessment of something
- Software (SW) measurement is the act of making a quantitative assessment of some software attributes, such as its complexity or reliability
- The long term goal of sw measurement is to use it to make judgment about sw quality or assess the effectiveness of sw processes, tools and methods
- A no of metrics can be used to assess the quality of a software product

# Reasons/Purposes for Software Measurement

(from Univ of Purdue & Others)

- **Projects often**
  - ✓ • Go over budget
  - ✓ • Miss deadlines
  - ✓ • Exhibit poor quality – does not satisfy all the functional and/or non functional requirements
- **Measurements can be used to improve the process and quality. / Systematic improvement is not possible without Measurement**
- ✓ Both short term (current) and long term (future)
- **You measure software product (or process) so as to evaluate / compare them with set standards / To ensure that industry standards and regulations are followed.**
- **Measurement Gives software products and processes a quantitative basis for evaluation**

# Software Measurement & Metrics

- **Software metrics**
- A metric is a standard for measuring or assessing something
- A sw metric is a measure of sw characteristics that are quantifiable or countable
- It is a measurement used to evaluate the effectiveness of a sw development process and the sw itself
- **Examples** of sw metrics are:
  - Size of a product in lines of code
  - Fox index – a measure of readability of a narratives' index
  - No of reported of faults in a delivered sw product
  - No of person-days required to develop the system completely
  - Etc

# Metrics Classification

- *Software metrics can be classified as*
- Process or Product metrics – univ of Purdue lecture note
  - **The software product includes the source code, executable code, and related documentation**
- Control or predictors Metrics – by Sommerville



# Metrics Classification

- **Process Metrics**
- Measurements of the development process
- Examples:
  - Development time
  - Methodology (Waterfall, Spiral, etc.)
  - Average years of experience for development staff
- **Product Metrics**

# Metrics Classification

- **Process Metrics**
- **Product Metrics**
- Measurements of software at any point in development
- Examples
  - Design Complexity
  - Program size (source or executable) in lines of code (LOC)
  - Pages of documentation
- etc

# Product Metrics

- **Size (product Size)**
  - Lines of code
  - File, Flow, Process (FFP) – Look for the meaning
  - Function points: **Based on the number of** inputs (Inp), outputs (Out), inquiries (Inq), master files (Maf), and interfaces (Inf)
- **Complexity (Product Complexity)**
- **Quality (Product Quality)**

# Product Metrics

- **Size (product Size)**
- **Complexity (Product Complexity)**
  - Software Science [Halstead]
  - Cyclomatic [McCabe]
  - Fan-in / Fan-out [Henry and Kafura]
  - Object-oriented [Morris]
- **Quality (Product Quality)**

# Product Metrics

- **Size (product Size)**
- **Complexity (Product Complexity)**
- **Quality (Product Quality)**
  - Defect counts
  - Reliability: mean time to failure
  - Maintainability: complexity, %comments

# Software Measurement & Metrics

- **Control Metrics (Sommerville)**
- Control metric – support process management
  - Are usually associated with sw process
  - E.g average effort & time required to repair reported defects
- There 3 types of control / Process metric:
  - Time taken for a particular process to be completed
  - The resources required for a particular process; e.g
    - Effort in person-days
    - Traveled costs
    - Computer resources (HW & SW)
  - The no of occurrences of a particular event; e,g

# Software Measurement & Metrics

- **Control Metrics contd**

- There 3 types of control / Process metric:
  - Time taken for a particular process to be completed
  - The resources required for a particular process; e.g
  - The no of occurrences of a particular event; e,g
    - No of defects discovered during code inspection
    - No of bugs reported in a delivered system
    - No of lines of code modified in response to a requirement change
- Process measurement is used by managers to decide if a process change should be made

# Software Measurement & Metrics

- **Predictor Metrics (Sommerville)**
- They help to predict characteristics/attributes of a sw product
- They are associated with the sw itself
- They are also called Product metrics
- Examples are:
  - Cyclomatic complexity of a program module
  - Average length of identifiers in a program,
  - etc
- Predictor metrics are used by managers to decide if software changes are necessary and if the sw is ready for release



# Software Measurement & Metrics

- **Product Metrics**

- They are predictor metrics and are used to quantify internal attributes of a sw system
- Examples:
  - System size, measured in lines of code or the number of methods associated with a each object
- They are in 2 categories : Dynamic & static metrics

- **Dynamic metrics**

- They are measurement made of a program in execution
- Examples
  - No of bugs reported
  - Run-time - The time to complete a computation
- Xc

# Software Measurement & Metrics

- **Static metrics of a sw product**
- These are collected by measurement made by representations of the system, such as the design, programs, documentation
- Examples are:
  - Fan-in/fan-out
  - Length of code
  - Cyclomatic complexity
  - Length of identifiers
  - Fox index
  - Depth of conditional nesting
- **Assignment:** Describe the meaning of those examples!; due in a week's time

# Software Measurement & Metrics

- **Highlights on Product metrics**
- Dynamic metrics help in assessing the efficiency & reliability of a system
- Static metrics assists in evaluating the complexity , understandability & maintainability of a system and it components
- The no of system failures and the type of failure is directly related to the reliability of the sw
- Program size, control complexity appear the most reliable predictors of understandability, system complexity & maintainability
- **Quiz:**
  - what is a software metric?
  - Describe different types of software metrics with example
  - Write comprehensively on Halstead's software metrics, Ref listing and citation in APA format. Due in a week's time

# Software Measurement & Metrics Contd

- Deliberately left blank

# Software Measurement & Metrics Contd

- **Examples of Static metrics of a sw product are:**
  - **Fan-in/fan-out**
    - Fan-in is the no of function/method that call another function method, (say X)
    - Fan-out the no of method called by function/method X.
    - High value fan-in implies that X is tightly coupled to the rest of design, thus changes to X will have extensive knock-out effect.
  - **Length of code**
    - This is a measure of the size of the program
    - Generally, the larger the size of the program ; the more complex and error-prone that component is likely to be
  - **Cyclomatic complexity**
  - **Length of identifiers**
  - **Fox index**
  - **Depth of conditional nesting**
- XX

# Software Measurement & Metrics Contd

- **Examples of Static metrics of a sw product are:**
  - **Fan-in/fan-out**
  - **Length of code**
  - **Cyclomatic complexity**
    - This is a measure of the control complexity of the system
    - The metric measures independent paths thru a source code.
    - Can informally be described as the no of decision points plus one.
    - McCabe (1976), the developer of the metric recommended that a program whose CC is more than 10 cyclometric should be split into smaller modules (Olatunji, 2019)
  - **Length of identifiers**
    - This is a measure of the average length of identifiers (names for variables, classes, methods, etc) in a program
    - The longer the identifier, the more likely they are to be meaningful & hence the more understandable the program
  - **Fox index**
  - **Depth of conditional nesting**
- XX

# Software Measurement & Metrics Contd

- Examples of Static metrics of a sw product are:
  - Fan-in/fan-out
  - Length of code
  - Cyclomatic complexity
  
  - Length of identifiers
  - Fox index
    - This is a measure of the average length of words and sentences in a document
    - The higher the value of the document's fog index, the more difficult to understand the document
  - Depth of conditional nesting
    - It is a measure of the depth of nesting of the if-statement in a program.
    - Deeply nested if-statement are hard to understand and potentially error-prone.
  
  - xx

