

*Software Process Implementation Guide for
Small-to-Medium Enterprises
Version 1.00 – December 2004*

Software Product Engineering

Eng. Madiha A. Hassan

December , 2004

© SECC, 2004

Software Product Engineering



Software Engineering
Competence Center

Software Product Engineering Process

Chapter 5

- Introduction
- Goals
- Policies
- Description
- Assumptions
- Sub-processes
 - Planning & initiation
 - Software design
 - Coding & Integration
 - Testing
 - Defect Tracking
 - Post - development

© SECC, 2004

Slide # 2

Software Product Engineering



Implementation Guidelines

Chapter 8

- Software Development Life Cycles Approach
- Traceability operations Guidelines
- Software Testing Guidelines



Roles & Responsibilities

Appendix A

- Product Manager
- Software Development Team (Including Testers)
- Technical Writer
- Project Manager, SQA, SCM, PR responsibilities



Metrics

Appendix B

- A simplified guide for data collection / analysis of defects
- Product Engineering Metrics.
 - Average number of defects / component
 - Average number of days to repair defects/ fault / errors (i.e. to close defect)
 -



SPE Templates

Appendix C

- Architectural Software Design Template
- Detailed Software Design Template
- Software Test Plan Template
- Test Design Specification Template
- Test Cases Specification Template
- Test Procedure Specification Template
- Software User Manual Template
- Software Transfer Document Template



SPE Checklists

Appendix D

- Software Design Checklist
- Software Testing Checklist
- User Documentation Checklist



Software Product Engineering Process

- Software product Engineering is concerned with a well - defined and integrated set of activities to produce correct, consistent software products effectively and efficiently
- SPE includes the technical activities producing the a software product, such as requirements engineering, design, coding generation, integration and testing



Software Product Engineering Process

- The engineering activities involve documenting software work products and maintaining traceability and consistency between them
- This process includes controlled transition between stages of the software life cycle and the activities needed to deliver high quality software product to the customer



Software Product Engineering Process

- A typical sub-processes of SPE are (Requirements Engineering, Design, Coding, Components integration, Testing , Operation and Maintenance)
- According to the scope of this project, the software requirement engineering will not be covered in detail in this PA (Requirements Engineering is combined with Requirements Management in RM PA)
- Integration with Requirements Engineering is described to ensure integrated life cycle of development



Software Product Engineering Process

- Integration with Requirement Engineering is assured and described all over the sub-processes and the starting point of SPE process in this course is the SRS
- Integration between sub-processes is assured through the traceability matrix updates
- Software operation & maintenance will be covered under (post-development process)



Why Written Processes ?

- Detailed and well written processes and procedures ensure that correct steps are being executed to facilitate a successful completion of a task
- This ensures that the process is repeatable



Why Standards & Templates ?

- This maintains document uniformity
- To facilitate communication between stakeholders of the process

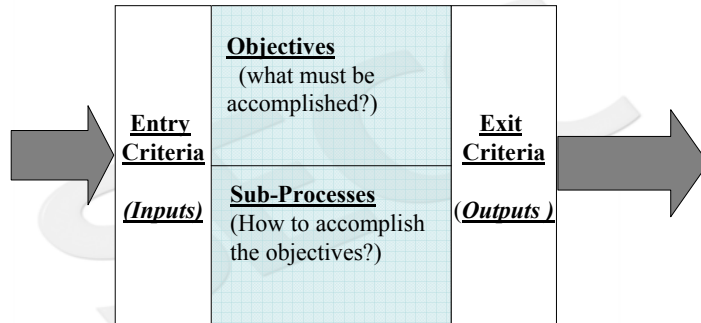


SPE Policies

- **Items to be addressed in the company policy document:**
 - Software products should be traceable to the SRS
 - Software product engineering activities should start as early as possible (i.e. the project initiation stage)
 -
 -



SPE Schematics



SPE Description

- An overview description for the process phases (sub-processes)
- SPE has 6 sub-processes
 - PE100 (Planning & Initiation)
 - PE200 (Software Design)
 - PE300 (Coding & Integration)
 - PE400 (Testing)
 - PE500 (Defect Tracking)
 - PE600 (Post-development)



SPE Assumptions

- The implementation should be coordinated by a solid and reasonable measurement program, published periodically.
- The early involvement of the software development team in the life cycle allows effective & efficient decisions which affect the quality of the final product
- The core processes (SQA, SCM & PR) should be considered in each sub-process
- Implementation is incremental with improvement iterations
- Requirement engineering is covered in the requirements management process and the main entry criteria for SPE is the SRS



Sub – Processes (Example) PE100 (Planning & Initiation)

Activities

- Sharing in selecting the appropriate SDLC
- Define a list of deliverables for each phase of the life cycle
- Estimate the life cycle resources & duration
- Define roles, responsibilities and allocate resources
- Identify the suitable software engineering methods & tools
- Identify risk factors and its possible mitigation plans (e.g. new technology risk,...)



Sub – Processes (Example) PE100 (Planning & Initiation)

Entry Criteria

- Allocated Budget
- Time Constrains

Exit Criteria

- Approved Software Development Plan

Inputs

- Project SOW

Output

- Initial Software Development Plan



Implementation guidelines Examples: Traceability Matrix Operations

1. Detailed description of traceability data which should be maintained (maintenance phase & source of data)
2. Basic techniques for handling traceability information (tables, lists, automated tools)



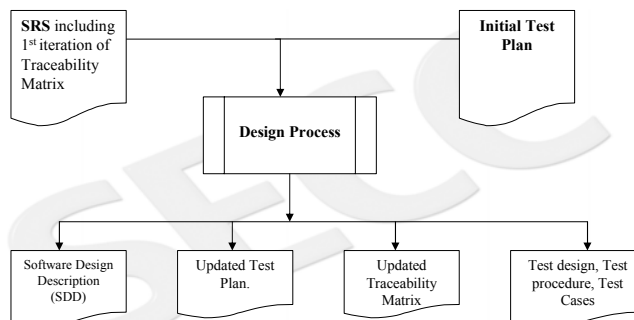
Implementation guidelines Examples : Software Testing guidelines

Based on IEEE std. 829- 1998

- **Test Preparation:**
 - Test plan: (describes how the testing will proceed)
 - Test design specifications: (describes what needs to be tested)
 - Test case specifications: (creates the test data to be running)
 - Test procedure: (describes how the tests are going to be running)
 - Test item transitional report: (specify the items released for testing)
- **Test Execution:**
 - Test log: (records the details of testing in time order)
 - Test incident report: (records the details of events failed)
- **Test Completion:**
 - Test summary report: (summarize testing)

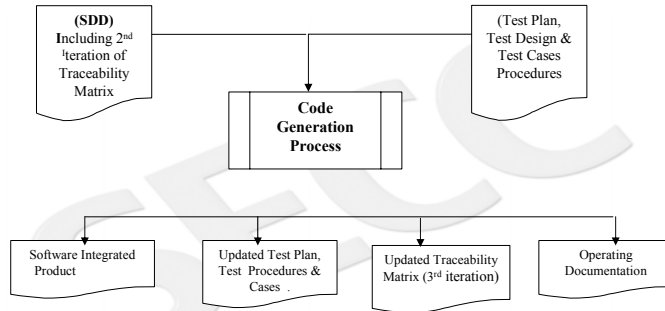


Ensure Integration Through Traceability Matrix

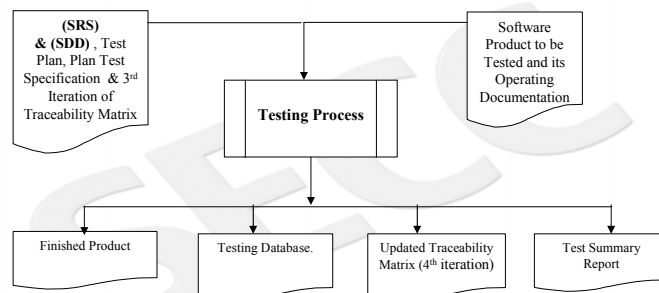




Ensure Integration Through Traceability Matrix



Ensure Integration Through Traceability Matrix



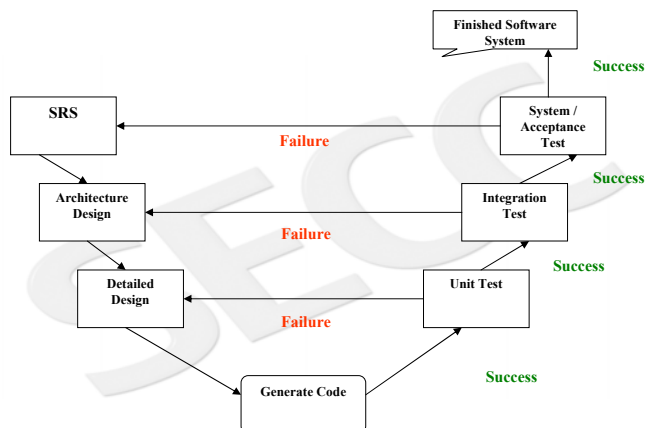


Testing Process Outputs & SDLC

	Unit testing	Integration testing	System Testing
Requirement Analysis		<ul style="list-style-type: none"> Test Plan 	<ul style="list-style-type: none"> Test Plan
Design	<ul style="list-style-type: none"> Test Plan Test Design Test Cases Test Procedure 	<ul style="list-style-type: none"> Test Design Test Cases Test Procedure 	<ul style="list-style-type: none"> Test Design Test Cases Test Procedure
Coding			
Testing	<ul style="list-style-type: none"> Test Execution Test Report 	<ul style="list-style-type: none"> Test Execution Test Report 	<ul style="list-style-type: none"> Test Execution Test Report

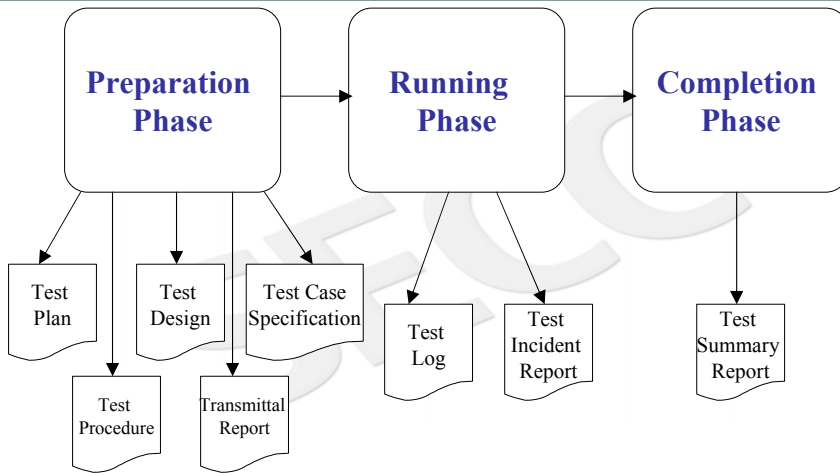


V. Model Application

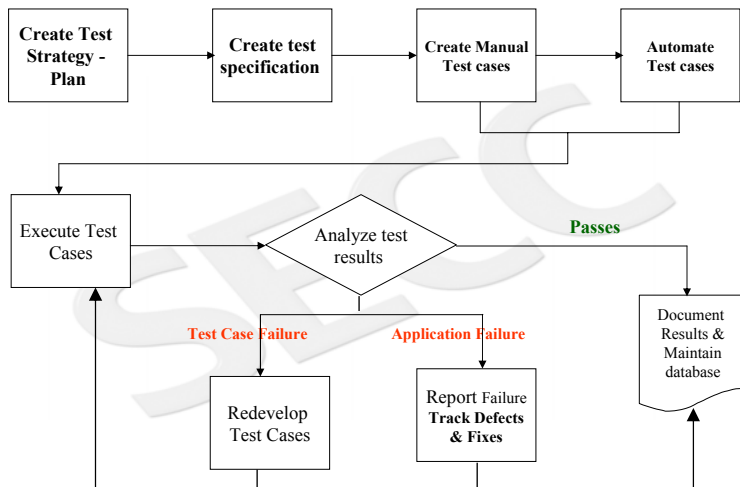




The Testing Process Output

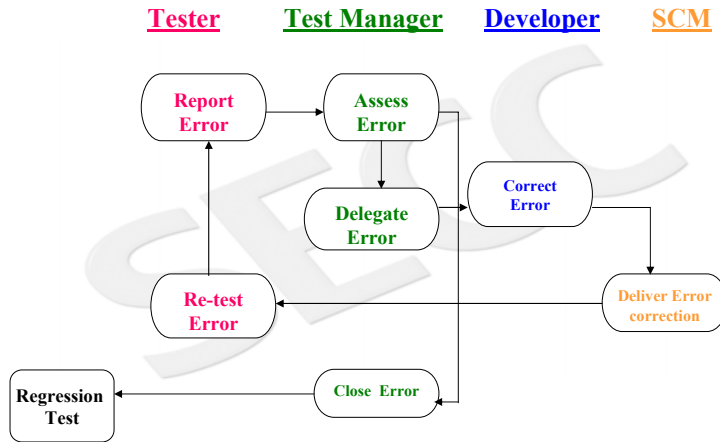


Testing Process Model for All Levels (Unit – Integration – System)





Error Tracking Procedure



Assets & References

- Customized Templates for SDD, Testing documentation, user documentation source is (IEEE 1016, IEEE 829, IEEE 1063-2001)
- Forms to facilitate deploying the processes
- Guidelines for implementation



Software Engineering
Competence Center

Software Product Engineering

Thank You