D50: Advances in Software Engineering

Case Study: Defining an Object-Oriented Development Process

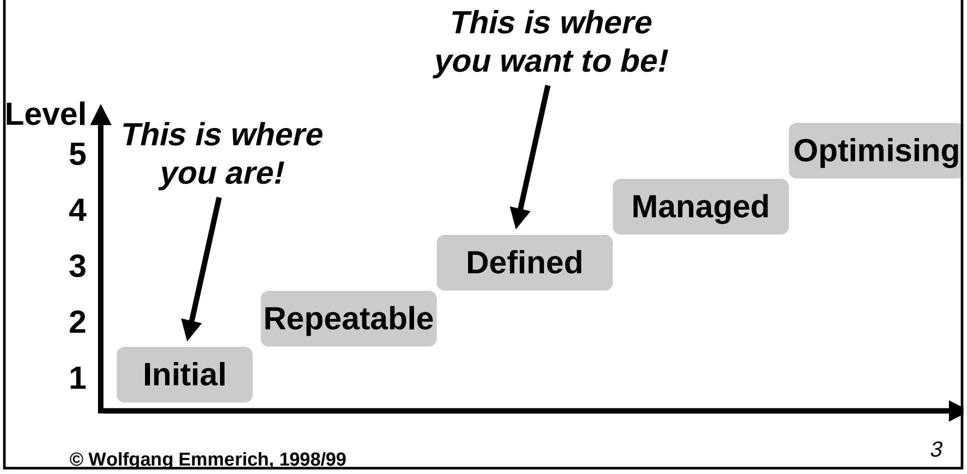
Wolfgang Emmerich

Why Define Process?

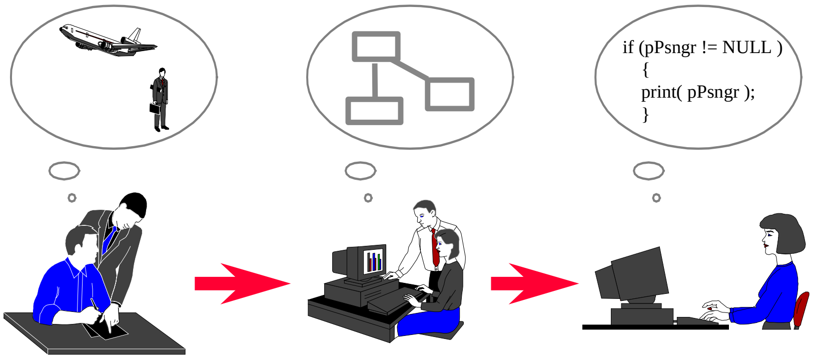
- Introduction of good practice
- Tailoring towards DG Bank & MRS/CAD
- Agreement on development process
- Basis for quality control
- Basis for process improvement
- Supports recruitment of internal / contracting of external developers
- Improves collaboration / embedding of contractors
- Records current experience for future use

Capability Maturity Model

- Developed by Watts Humphrey at the SEI
- US DoD requires Level 3 maturity of contractors

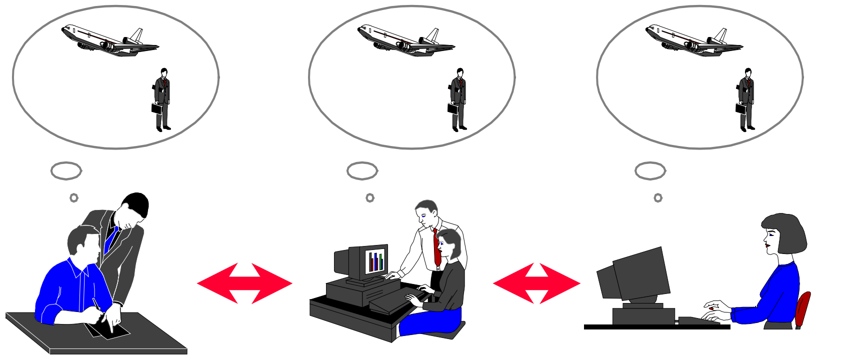


Conventional SW Engineering



```
if (pPsngr != NULL )
   {
   print( pPsngr );
   }
```

Object-Oriented SW Engineering



Users

Analyst/Designers

Programmers

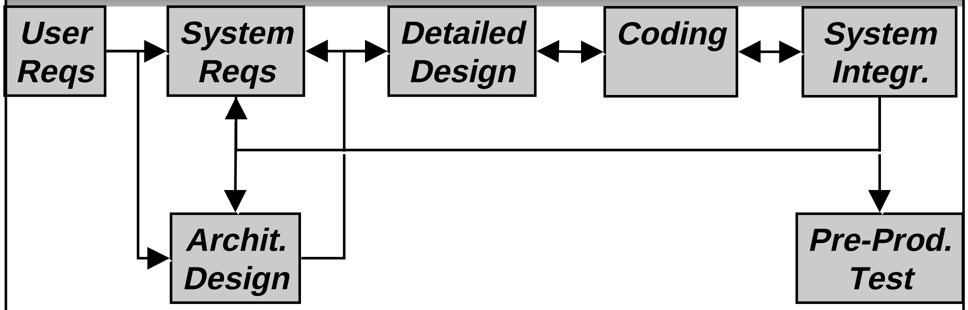
History

- Draft 2 written by N. Schmitt and W. Schwarz
- One-day Workshop organised by LogOn
- Briefing sessions with
  - Ms. Cruz
  - Mr. Büchler
  - Mr. Mohr
  - Mr. Köhling
  - Mr. Wolter
  - Mr. Schwarz
  - Mr. Knauf
  - Mr. Bosselmann
  - Mr. Wolf

What needs to be defined?

- ***Activities***
  - ***Leader***
  - ***Reviewers***
  - ***Resulting Deliverables***
  - ***Dependencies to/from other activities***
- ***Deliverables***
  - ***Purpose***
  - ***Content***
  - ***Owner***
  - ***Tools to produce them***
- ***Roles***
  - ***Reporting structure***
  - ***Responsibilities***
  - ***Skills***

Stages of Development Process

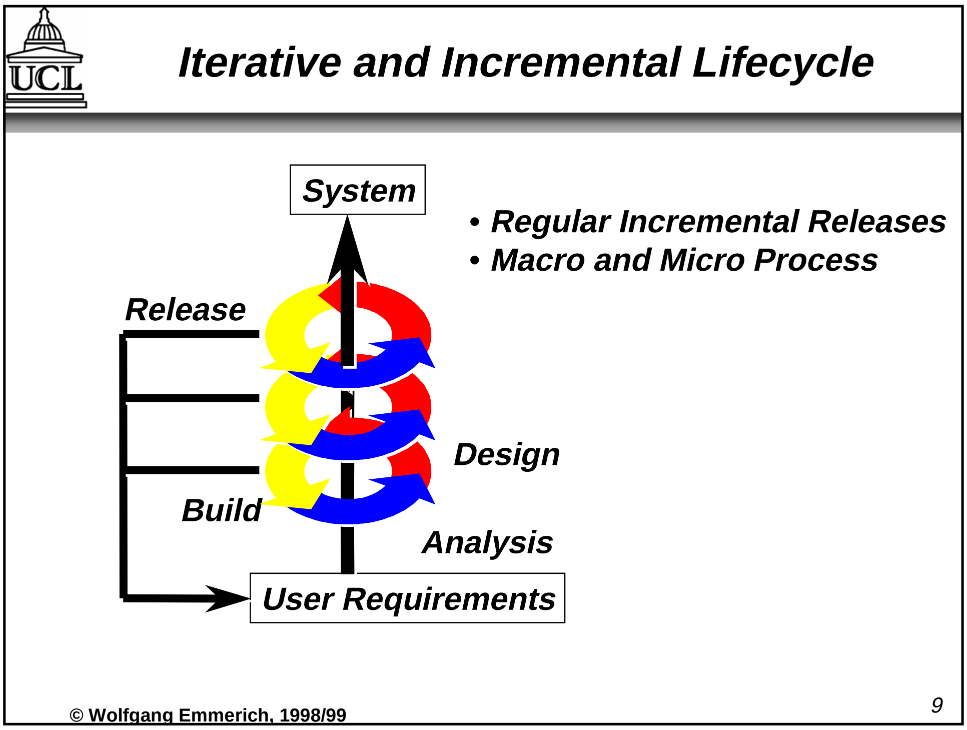


Project and Risk Management

System Management

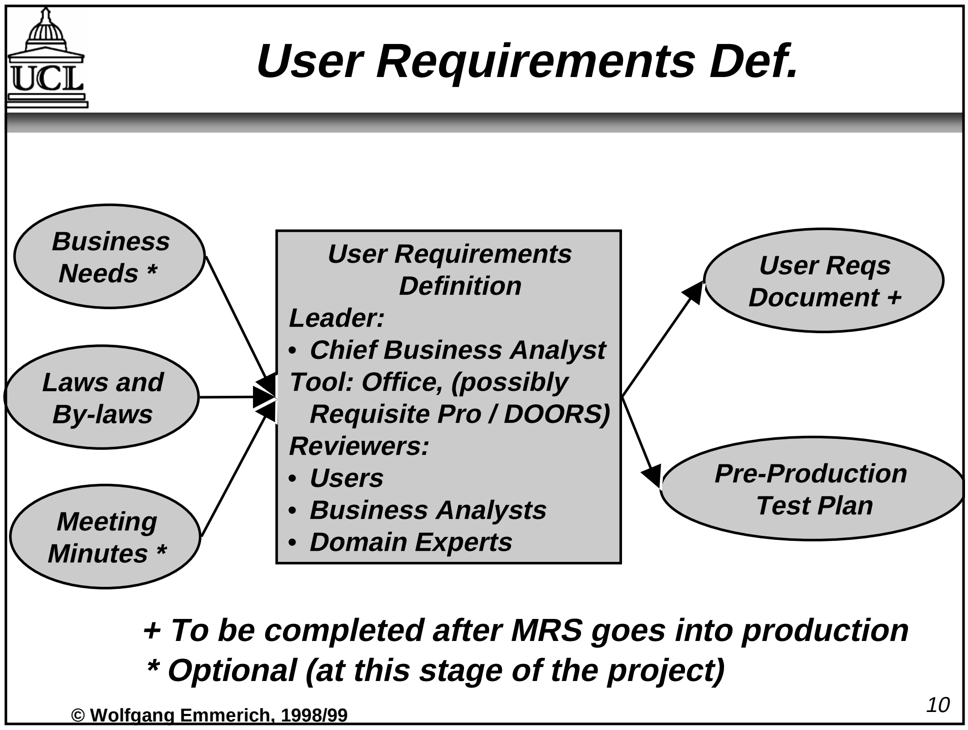
Version and Configuration Management

Quality Management



Iterative and Incremental Lifecycle

- Regular Incremental Releases
- Macro and Micro Process



User Requirements Def.

User Requirements Definition

Leader:

- Chief Business Analyst

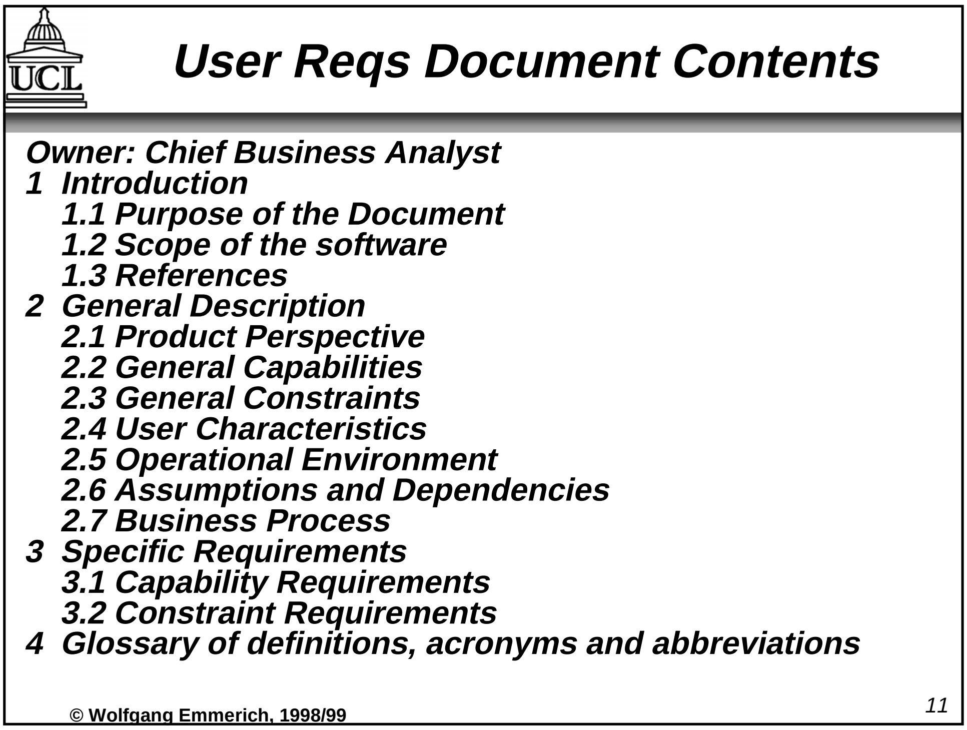
Tool: Office, (possibly Requisite Pro / DOORS)

Reviewers:

- Users
- Business Analysts
- Domain Experts

+ To be completed after MRS goes into production

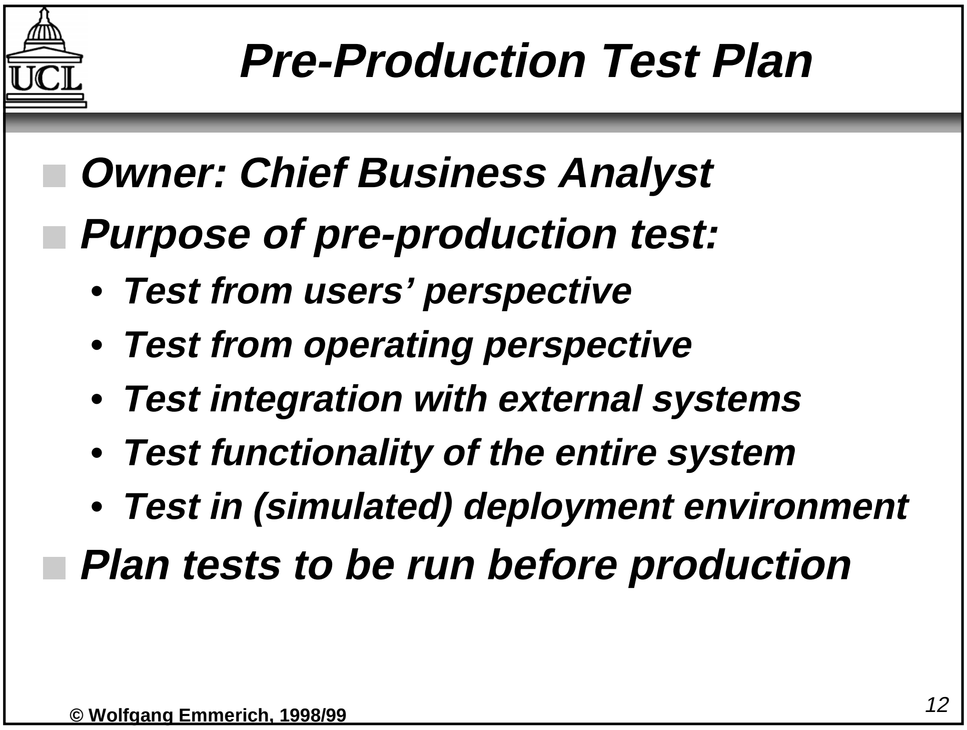
* Optional (at this stage of the project)



User Reqs Document Contents

Owner: Chief Business Analyst

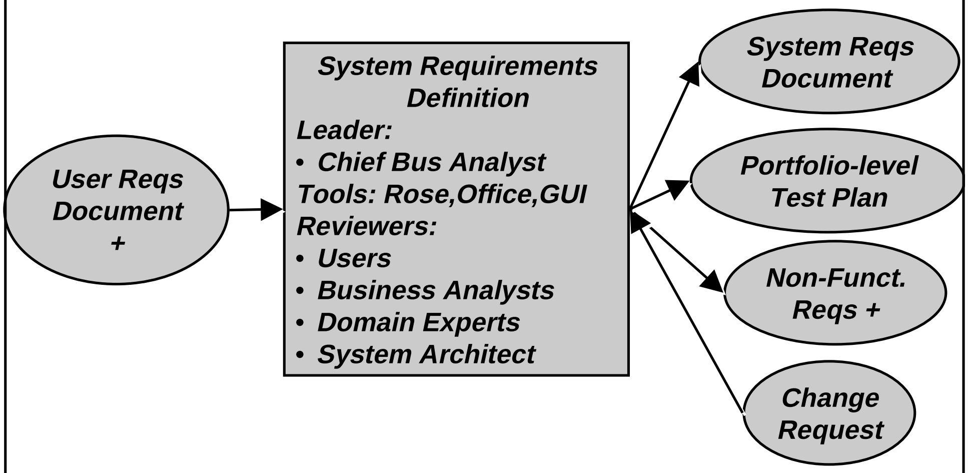
1 Introduction
  1.1 Purpose of the Document
  1.2 Scope of the software
  1.3 References
2 General Description
  2.1 Product Perspective
  2.2 General Capabilities
  2.3 General Constraints
  2.4 User Characteristics
  2.5 Operational Environment
  2.6 Assumptions and Dependencies
  2.7 Business Process
3 Specific Requirements
  3.1 Capability Requirements
  3.2 Constraint Requirements
4 Glossary of definitions, acronyms and abbreviations



Pre-Production Test Plan

- Owner: Chief Business Analyst
- Purpose of pre-production test:
  - Test from users' perspective
  - Test from operating perspective
  - Test integration with external systems
  - Test functionality of the entire system
  - Test in (simulated) deployment environment
- Plan tests to be run before production

System Requirements Def.



System Reqs Document

Owner: Chief Business Analyst

1 UML Use Case Model (Rose) +
2 UML Interaction Diagrams (Rose) +
- UML Sequence Diagrams
- UML Collaboration Diagrams

3 UML Analysis Class Diagram (Rose) +

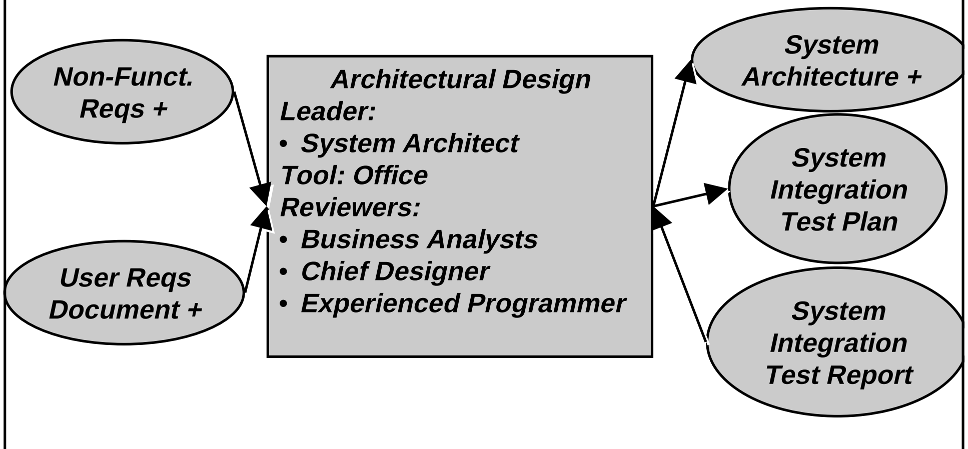
Analysis Document

4 User Interface Prototypes (GUI Tools, Office) *
5 Specification of Report Format (Office)
6 User Manual (Office)
- User Reference Manual
- User’s Guide

Non-Functional Requirements

- *Owner: Chief Business Analyst*
- *Capacity Requirements*
- *Performance Requirements*
- *Accuracy Requirements*
- *Constraint Requirements*
  - *Communication Interfaces*
  - *Hardware Interfaces*
  - *Interfaces to external software components*
  - *HCI Requirements*
  - *Adaptability Requirements*
  - *Availability Requirements*
  - *Portability Requirements*
  - *Security Requirements*
  - *Compliance to Standards*
  - *Resources*
  - *Time scales*

Architectural Design



Architectural Design
Leader:
• System Architect
Tool: Office
Reviewers:
• Business Analysts
• Chief Designer
• Experienced Programmer

System Architecture

Owner: System Architect

1 Architecture Overview and Rationale

2 Services provided by Modules

3 Resources used by Modules
(disk, memory, CPU time, real-time)

4 Dependencies between Modules

- *Control flow*
- *Data flow*
- *Synchronization*

5 Rationale of chosen implementation languages

6 Traceability to non-functional requirements?

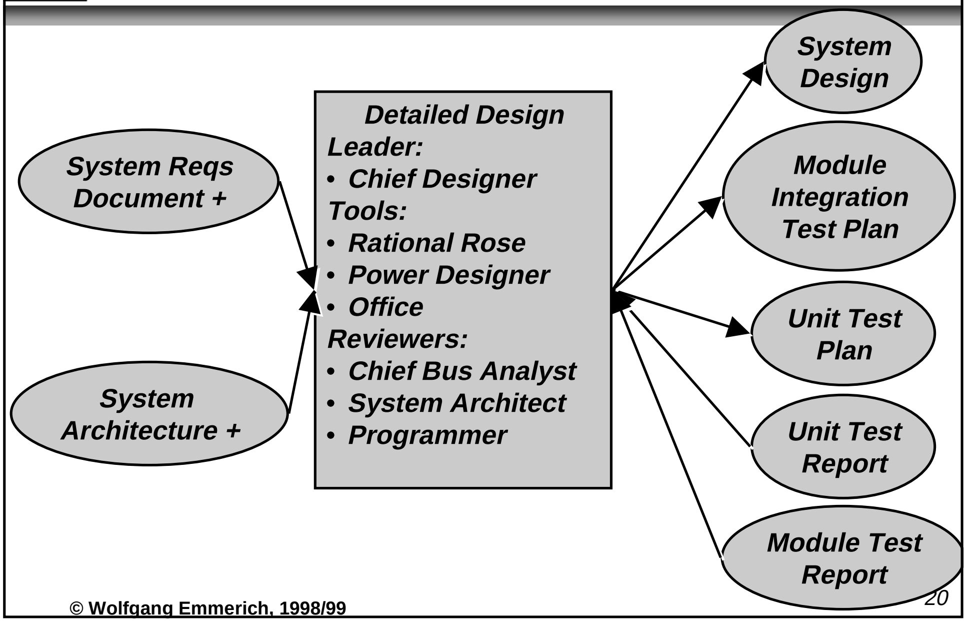
Portfolio-level Test Plan

- *Owner: Chief Business Analyst*
- *Purpose of portfolio-level test*
  - *Tests correctness of Value at Risk Figures based on trade portfolios*
  - *Compare Exact Pricing VaR computed by MRS/CAD and MRS/CAD light (spreadsheet)*
  - *Test hierarchical simulation and variance/covariance VaR by comparison to Exact Pricing*
  - *Test Back-testing and stress-testing*
- *Plan how to perform portfolio-level test*

System Integration Test Plan

- Owner: Test Manager
- Plan how to test integrated system
- Purpose of System Integration Test:
  - Validate that modules work together
  - Trace how trades are processed through the system
  - Validate that non-functional requirements are met

Detailed Design



Detailed Design

Leader:

- Chief Designer

Tools:

- Rational Rose
- Power Designer
- Office

Reviewers:

- Chief Bus Analyst
- System Architect
- Programmer

Module Integration Test Plan

- ***Owner: Test Coordinator of Module***
- ***Plan how to perform module test***
- ***Plan is required for modules***
  - ***GDB***
  - ***MRS***
- ***Purpose of module integration test:***
  - ***Test the integration of packages that comprise a module***
  - ***Meet functionality and technical requirements imposed by design***
  - ***Test-Coordinator decides when hand over module to other teams***

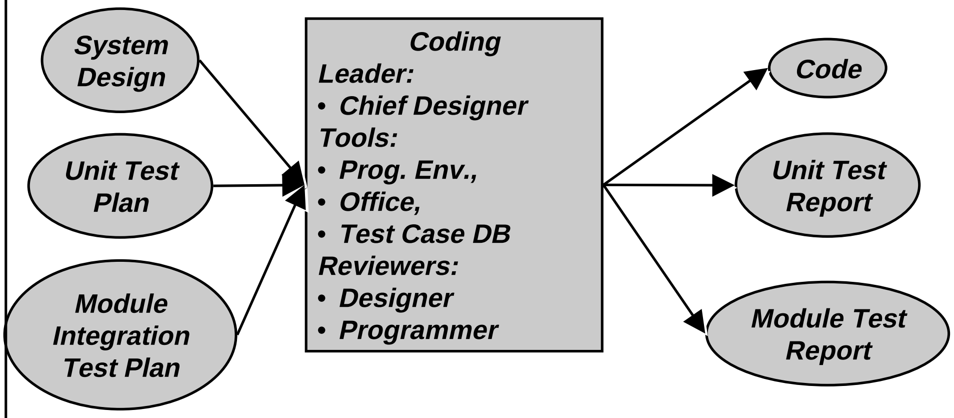
Unit Test Plan

- ***Owner: Test Coordinator of resp. Module***
- ***Plan how to perform unit tests***
- ***Plan is required for each unit (UML package in MRS and GDB, classes in SAI)***
- ***Purpose of Unit Test***
  - ***Black/Box Test in MRS and GDB***
  - ***White/Box Test in SAI***
  - ***Test functionality exported by unit***
  - ***Test functionality and behaviour of design***
  - ***Test behaviour with invalid input parameters***
  - ***Test coordinator decides when to hand over unit to other teams***

System Design

- Owner: Chief Designer
- Design Overview
- Detailed Design: UML Class Diagrams (Rose) and Design Paper (Soda)
  - Broken down into packages
  - Responsibility delegation for each package
- UML State Diagrams for critical classes (Rose)
- UML Interaction Diagrams for critical scenarios (Rose)
  - UML Sequence Diagrams
  - UML Collaboration Diagrams
- UML Component Diagram (Rose)
- Data Model (Power Designer)

Coding



Coding

Leader:
- Chief Designer

Tools:
- Prog. Env.,
- Office,
- Test Case DB

Reviewers:
- Designer
- Programmer

Code

Owner: Responsible for Package

1 Implementations of Classes

- ***C++ Code***
- ***Documentation***
- ***Rationale for Algorithms chosen***

2 SQL schemas

3 Makefiles

4 Scripts, libraries, executables, configfiles

Unit Test Report

Owner: Test Coordinator of Module

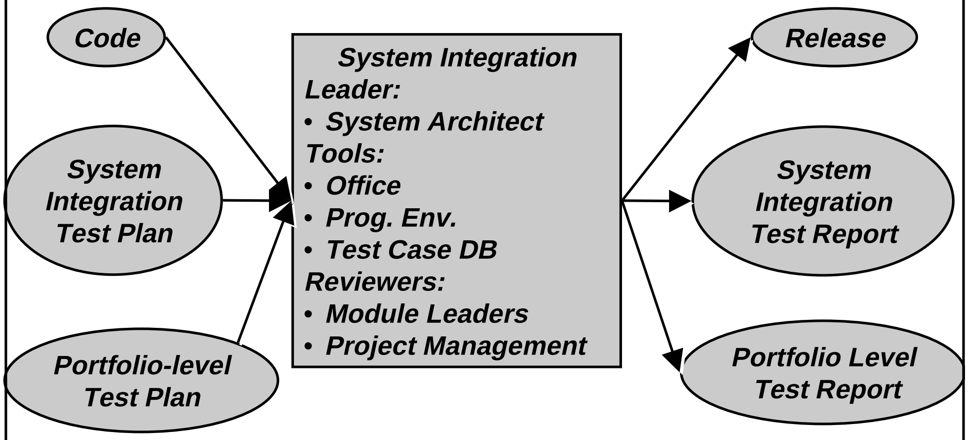
1 Test Plan (From Detailed Design)

2 Test Architecture

3 Test Case Description

4 Test Reports

System Integration



Code

System Integration Test Plan

Portfolio-level Test Plan

System Integration
Leader:
- System Architect

Tools:
- Office
- Prog. Env.
- Test Case DB

Reviewers:
- Module Leaders
- Project Management

Release

System Integration Test Report

Portfolio Level Test Report

Release

- Owner: System Architect
- Executable Increment
- Release Notes
- Package for Release Installation
- Installation Manual

System Integration Test Report

Owner: Test Manager

1 Test Plan (from Architectural Design)

2 Test Architecture

3 Test Case Descriptions

4 Test Reports

Portfolio-level Test Report

Owner: Chief Business Analyst

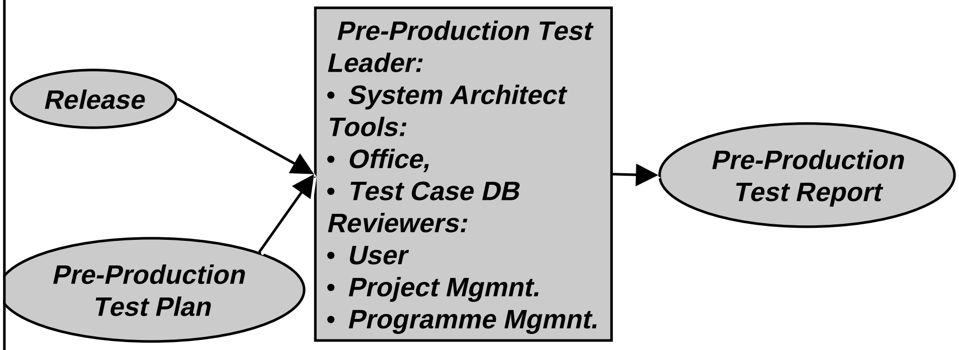
1 Test Plan (from System Requirements)

2 Test Architecture

3 Test Case Descriptions

4 Test Reports

Pre-Production Test



Release

Pre-Production Test Plan

Pre-Production Test

Leader:

- System Architect

Tools:

- Office,
- Test Case DB

Reviewers:

- User
- Project Mgmnt.
- Programme Mgmnt.

Pre-Production Test Report

Pre-Production Test Report

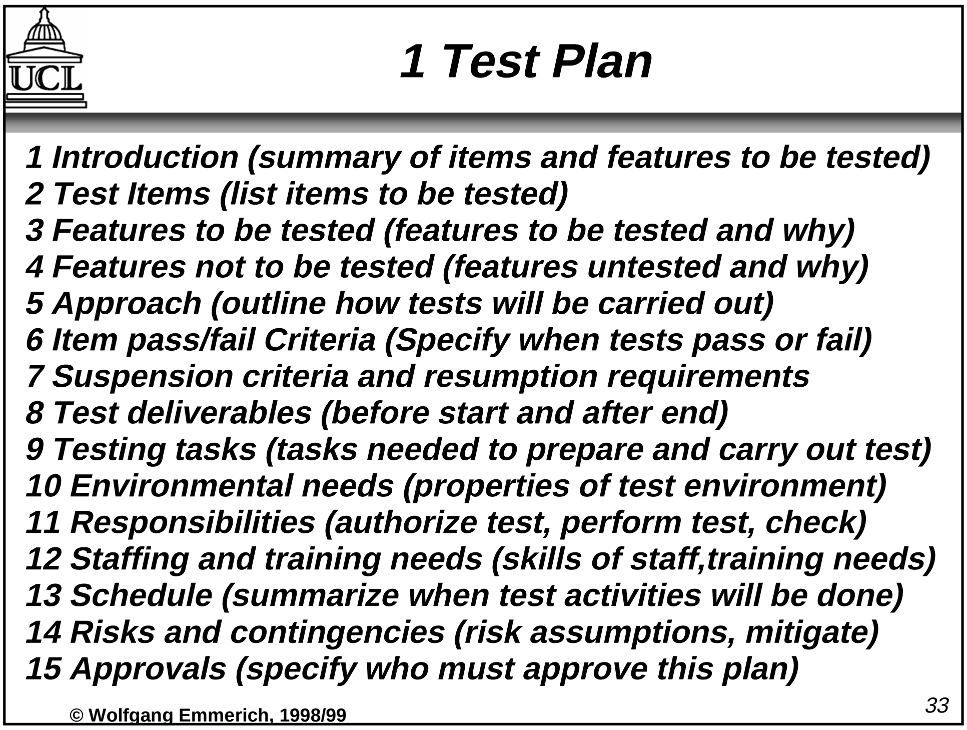
Owner: System Architect

1 Test Plan (from User Requirements)

2 Test Architecture

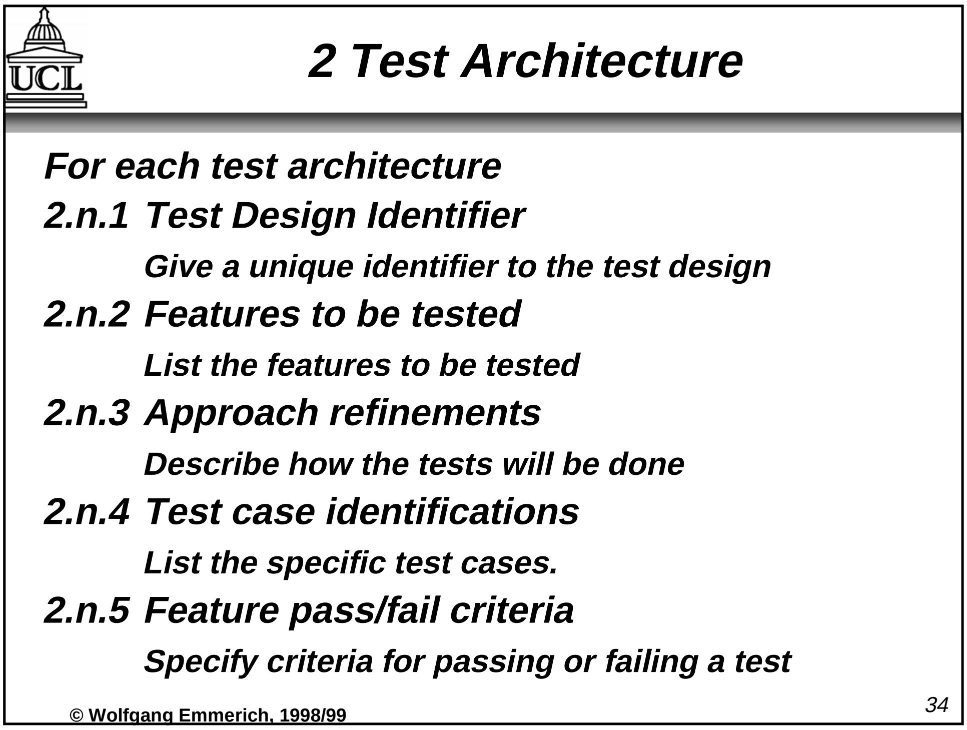
3 Test Case Descriptions

4 Test Reports



1 Test Plan

1 Introduction (summary of items and features to be tested)
2 Test Items (list items to be tested)
3 Features to be tested (features to be tested and why)
4 Features not to be tested (features untested and why)
5 Approach (outline how tests will be carried out)
6 Item pass/fail Criteria (Specify when tests pass or fail)
7 Suspension criteria and resumption requirements
8 Test deliverables (before start and after end)
9 Testing tasks (tasks needed to prepare and carry out test)
10 Environmental needs (properties of test environment)
11 Responsibilities (authorize test, perform test, check)
12 Staffing and training needs (skills of staff,training needs)
13 Schedule (summarize when test activities will be done)
14 Risks and contingencies (risk assumptions, mitigate)
15 Approvals (specify who must approve this plan)



2 Test Architecture

For each test architecture

2.n.1 Test Design Identifier

Give a unique identifier to the test design

2.n.2 Features to be tested

List the features to be tested

2.n.3 Approach refinements

Describe how the tests will be done

2.n.4 Test case identifications

List the specific test cases.

2.n.5 Feature pass/fail criteria

Specify criteria for passing or failing a test

3 Test Case Specification

For each test case:

3.n.1 Test case identifier

Give a unique identifier for the case

3.n.2 Test items

List the items to be tested

3.n.3 Input specifications

Describe the input for the case

3.n.4 Output specifications

Describe the output required from case

3.n.5 Environmental needs

Describe the test environment

3.n.6 Special procedural requirements

Describe special constraints on this case

3.n.7 Inter case dependencies

Test cases that must precede this case

4 Test Reports

For each execution of a test procedure

4.n.1 Test report identifier

Give a unique identifier for the test report

4.n.2 Description

List the items being tested

4.n.3 Activity and event entries

Identify the test procedure.

Say when the test was done, who did it and who witnessed it.

Describe the environmental conditions.

Describe what happened.

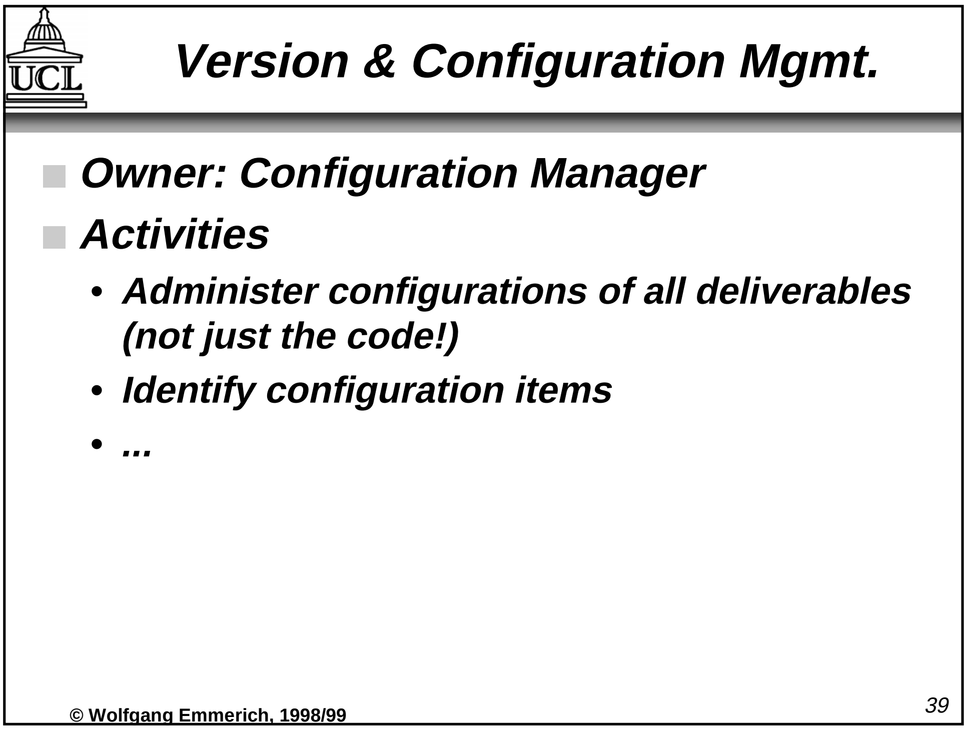
Describe where the outputs of the test procedure are kept.

Project & Risk Management

- Owner: Project Manager
- Activities:
  - Organising the Project
    - define roles of team members
    - define team structure
    - document responsibilities of team members
  - Risk Management
    - quality and stability of user requirements
    - level of definition & stability of external interfaces
    - adequacy and availability of resources
    - availability and quality of tools
    - staff training and experience
    - definition of responsibilities
    - short time scales
    - technical novelty of the project

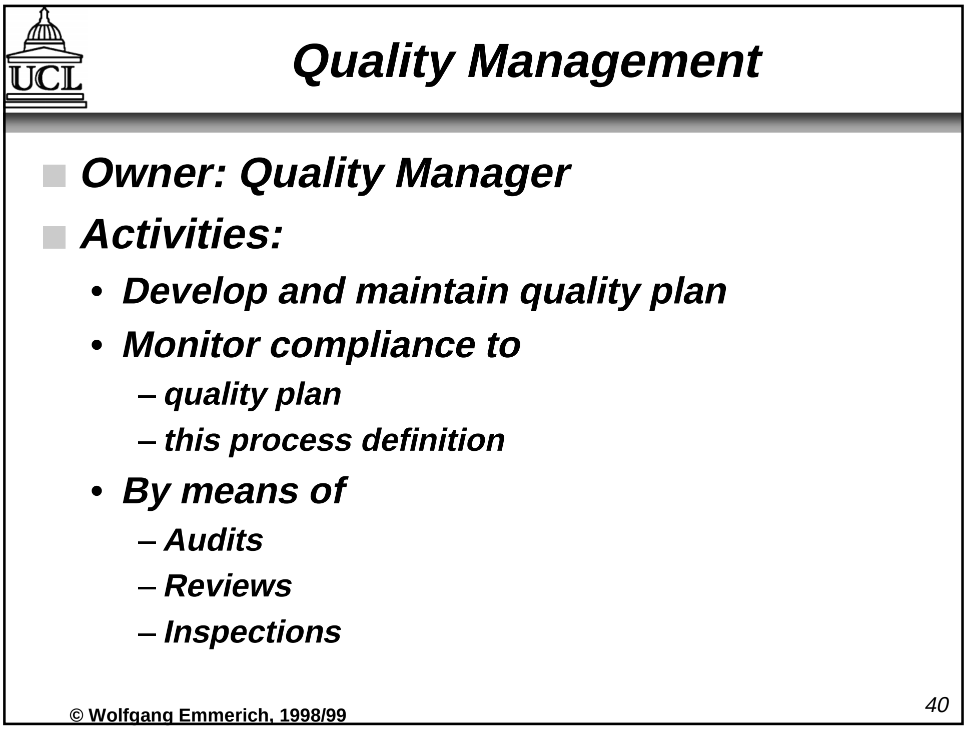
System Management

- Owner: System Manager
- Activities:
  - Maintain Hardware/Software Platform
  - Maintain Network
  - Maintain Tool Infrastructures
  - Help desk
  - Administer Databases
  - Administer and Backup File stores
  - Administer Developer Accounts



Version & Configuration Mgmt.

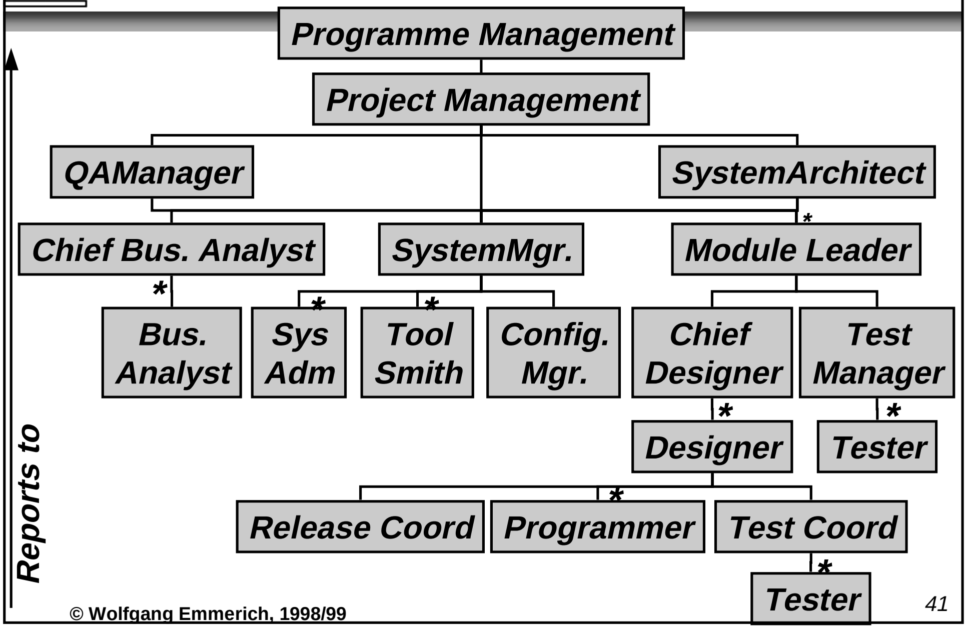
- Owner: Configuration Manager
- Activities
  - Administer configurations of all deliverables (not just the code!)
  - Identify configuration items
  - ...



Quality Management

- Owner: Quality Manager
- Activities:
  - Develop and maintain quality plan
  - Monitor compliance to
    - quality plan
    - this process definition
  - By means of
    - Audits
    - Reviews
    - Inspections

Roles



Programme Management

Project Management

QAManager

SystemArchitect

Chief Bus. Analyst

SystemMgr.

Module Leader

Bus. Analyst

Sys Adm

Tool Smith

Config. Mgr.

Chief Designer

Test Manager

Designer

Tester

Release Coord

Programmer

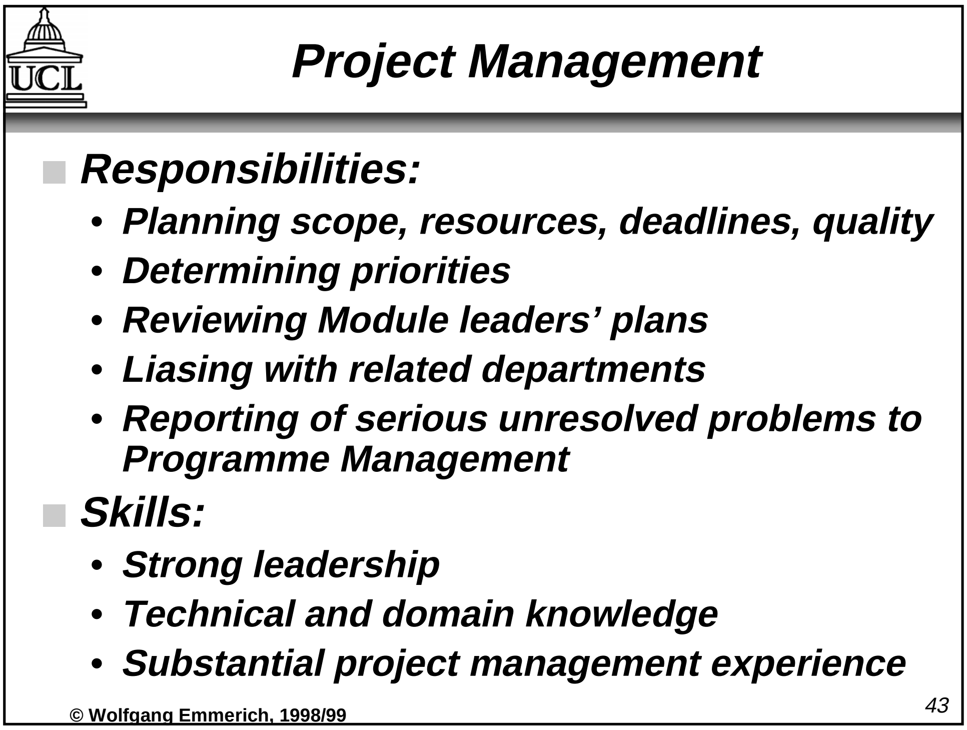
Test Coord

Tester

Reports to

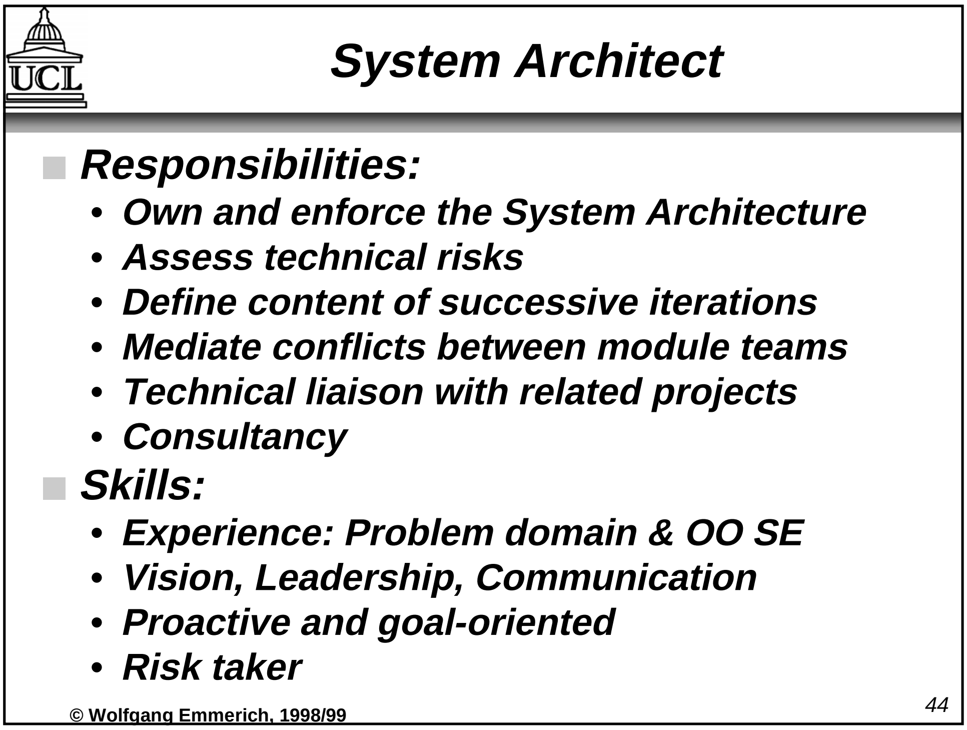
Programme Management

- Responsibilities
  - Agree on project aims and objectives
  - Agree on project plan (schedule, costs, quality, resources)
  - Receive monthly reports from project management
  - Decide on staffing resources
  - Agree / Reject changes to project plan
- Skills
  - Very strong leadership
  - Diplomacy
  - Communication skills



Project Management

- **Responsibilities:**
  - *Planning scope, resources, deadlines, quality*
  - *Determining priorities*
  - *Reviewing Module leaders' plans*
  - *Liasing with related departments*
  - *Reporting of serious unresolved problems to Programme Management*
- **Skills:**
  - *Strong leadership*
  - *Technical and domain knowledge*
  - *Substantial project management experience*



System Architect

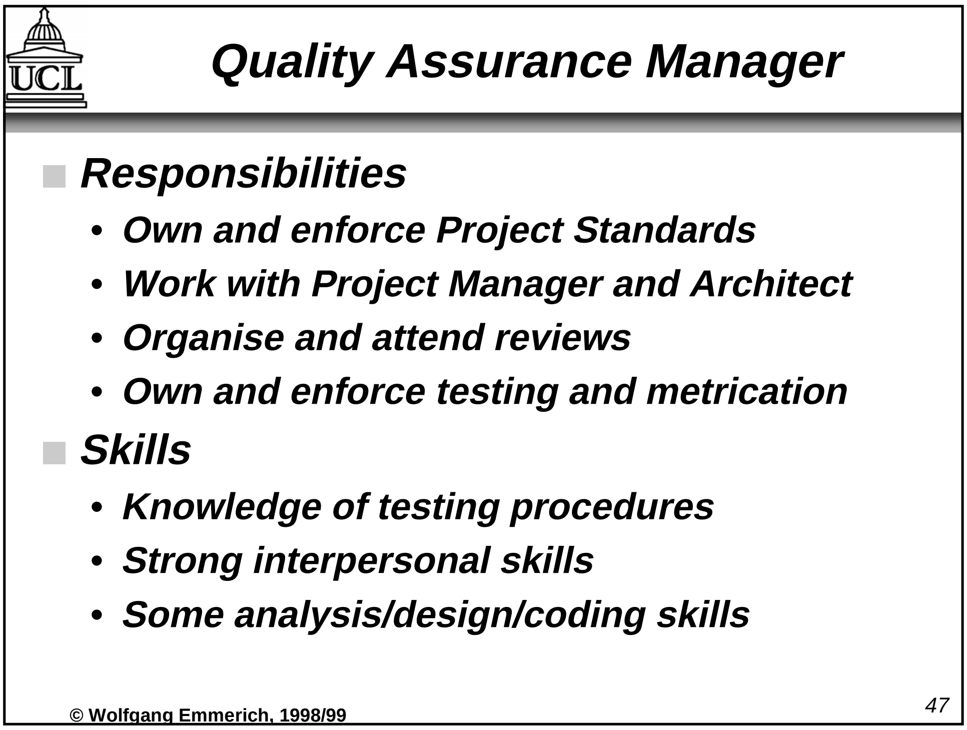
- **Responsibilities:**
  - *Own and enforce the System Architecture*
  - *Assess technical risks*
  - *Define content of successive iterations*
  - *Mediate conflicts between module teams*
  - *Technical liaison with related projects*
  - *Consultancy*
- **Skills:**
  - *Experience: Problem domain & OO SE*
  - *Vision, Leadership, Communication*
  - *Proactive and goal-oriented*
  - *Risk taker*

Chief Business Analyst

- ***Responsibilities:***
  - ***Advise and support the Architect***
  - ***Mentor and lead Business Analysts***
  - ***Arrange and attend Analysis reviews***
  - ***Define key interfaces***
- ***Skills***
  - ***Leadership: manage Business Analysts***
  - ***Communication***
  - ***Proactive and goal oriented***
  - ***Strong analysis skills and domain knowledge***
  - ***Experience in testing business functionality***

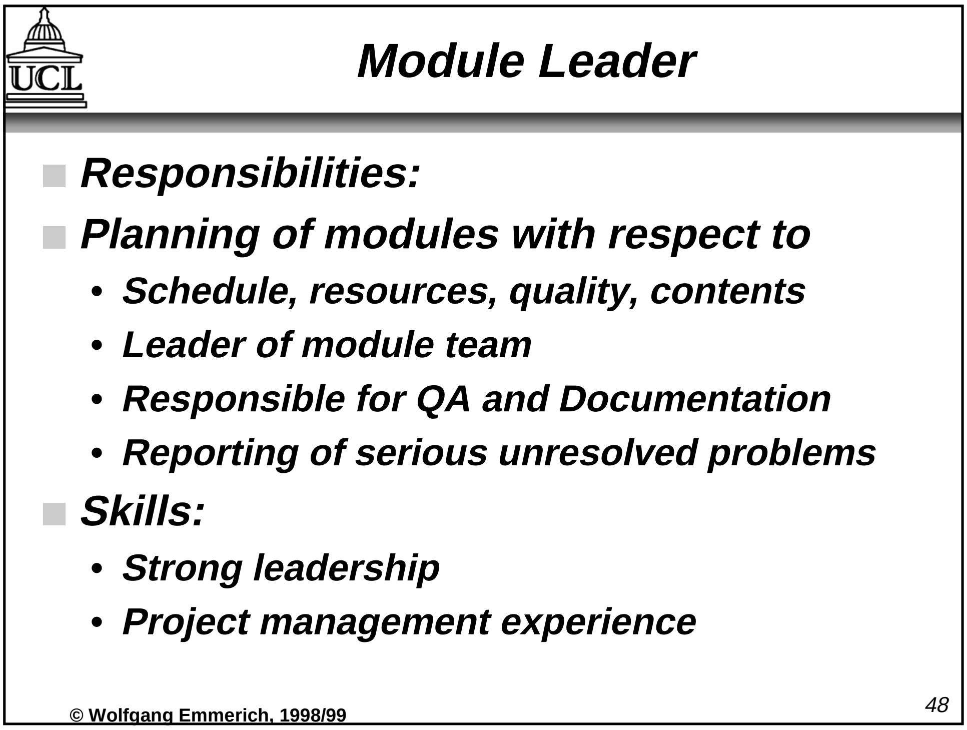
Business Analyst

- ***Responsibilities:***
  - ***Work with Chief Business Analyst***
  - ***Define interfaces and identify classes, packages, patterns***
  - ***Testing***
- ***Skills***
  - ***Communication skills***
  - ***Excellent analysis skills (finding abstractions)***
  - ***Excellent business domain knowledge***
  - ***Some programming skills***



Quality Assurance Manager

- **_Responsibilities_**
  - ***Own and enforce Project Standards***
  - ***Work with Project Manager and Architect***
  - ***Organise and attend reviews***
  - ***Own and enforce testing and metrication***
- **_Skills_**
  - ***Knowledge of testing procedures***
  - ***Strong interpersonal skills***
  - ***Some analysis/design/coding skills***



Module Leader

- **_Responsibilities:_**
- **_Planning of modules with respect to_**
  - ***Schedule, resources, quality, contents***
  - ***Leader of module team***
  - ***Responsible for QA and Documentation***
  - ***Reporting of serious unresolved problems***
- **_Skills:_**
  - ***Strong leadership***
  - ***Project management experience***

Configuration Manager

- Responsibilities
  - Owns set up and operation of the whole CM environment
  - Ensures CM standards are adhered to
  - Provides CM training/tools to team members
- Skills
  - Detailed knowledge of the CM system
  - Good communication skills
  - Risk aversive
  - Reliable and trustworthy

Toolsmith

- Responsibilities
  - Buy or build the best software tools to maximise team productivity
- Skills
  - Intimate knowledge of available tools
  - Good communication skills
  - Excellent coding Skills
  - Inventive, proactive and goal-oriented
  - Likes to tinker

Chief Designer

- **Responsibilities**
  - *Manage team of Designers*
  - *Co-ordinate all design activities*
  - *Work with System Architect and Chief Business Analyst to keep design and analysis models in step*
- **Skills**
  - *Leadership: manage Designers*
  - *Good communication skills*
  - *Proactive and goal oriented*
  - *Strong design and programming skills*

Designer

- **Responsibilities**
  - *Mentor and lead 3 to 4 programmers*
  - *Detailed class design*
  - *Owns one or more packages from detailed design through to implementation*
- **Skills**
  - *Leadership of small teams*
  - *Good communication skills*
  - *Excellent design and programming skills*

Test Manager

- Responsibilities
  - Creates test plan for system testing
  - Creates testing standards and works with QA Manager to enforce them
  - Manages a team of testers
- Skills
  - Leadership
  - Excellent communication skills
  - Experience in testing

Tester

- Responsibilities
  - Unit and system testing
  - Designing test strategies
  - Track and document all tests
- Skills
  - Knowledge of testing
  - Programming skills

Programmer

- Responsibilities
  - Implement the design model
  - Tactical class design
  - Class-level testing
  - Participate in code walkthroughs
- Skills
  - Good coding skills and likes to code!
  - Familiar with basic OOA/OOD principles
  - Understands the modelling language
  - Perhaps has a specialisation e.g. GUI

Release Coordinator

- Responsibilities
  - Works with Users, Project Manager and System Architect to create a release plan
  - Monitors and maintains the plan
  - Advises the project manager on any non-technical risks related to the release plan
- Skills
  - Good communication and interpersonal skills
  - Good domain knowledge
  - Project planning skills

Test Coordinator

- *Responsibilities*
  - *Create test plan for module and unit tests*
  - *Works with QA manager to enforce test plans*
  - *May manage more testers*
- *Skills*
  - *Excellent communication skills*
  - *Experience in testing*

Summary of Benefits

- *Introduce proven process into MRS*
- *Indicate documentation contents as basis for quality control*
- *Assist in identifying problems/risks early*
- *Guide staffing of development team*
- *Identify how to migrate to this process*
- *Record current experience for future use*