Software Project Management

Course Contents:

Unit 03: Techniques of Planning, Controlling and Automating Software Process

3.1 Iterative Process Planning (Process Work Breakdown Structure, Planning Guidelines, Cost and Schedule Estimation Process, Iteration Planning Process)

3.2 Project Organization and Responsibilities

3.3 Process Automation – Tools and Environment

3.4 Project Control and Process Automation

3.5 Process Customization

- **Iterative Process Planning**

➢ Work Breakdown Structures

➢ Planning Guidelines

➢ The Cost and Schedule Estimating Process

➢ The Iteration Planning Process

➢ Pragmatic Planning

- **Project Organizations and Responsibilities**

➢ Line-of-Business organizations

➢ Project Organizations

➢ Evolution Organizations

- **Process Automation**

➢ Tools: Automation Building Blocks

➢ The Project Environment

- **Project Control and Process Instrumentation**
  - ➢ The Seven Core Metrics
  - ➢ Management Indicators
  - ➢ Quality Indicators
  - ➢ Life-Cycle Expectations
  - ➢ Pragmatic Software Metrics
  - ➢ Metrics Automation
- **Process Atomization**
  - ➢ Process Discriminants
  - ➢ Example: Small-Scale Project Versus Large-scale Project

- **Iterative Process Planning**
- ➢ Work Breakdown Structures
- ➢ Planning Guidelines
- ➢ The Cost and Schedule Estimating Process
- ➢ The Iteration Planning Process
- ➢ Pragmatic Planning

work breakdown structure (WBS)

- A **work breakdown structure** (**WBS**) is a key project deliverable that organizes the team's **work** into manageable sections.

- The **work breakdown structure** visually defines the scope into manageable chunks that a project team can understand, as each level of the **work breakdown structure** provides further definition and detail.

- The development of a work breakdown structure is dependent on the project management style, organizational culture, customer preference, financial constraints and several other hard-to-define parameters.

- A WBS is simply a hierarchy of elements that decomposes the project plan into the discrete work tasks.

- A WBS provides the following information
  - A delineation of all significant work
  - A clear task decomposition for assignment of responsibilities
  - A framework for scheduling, budgeting, and expenditure
    tracking.

What is process breakdown structure?

•Product **Breakdown Structure** (BBS) is a project management tool and important part of the project planning.

•The product **breakdown structure** defines subtasks or work packages and describes the relationship between work packages.

•This **process** helps to organize the projects gut and to define the project frameworks.

Why use a WBS in project management?

- Estimate the cost of a project.
- Establish dependencies.
- Determine a project timeline and develop a schedule.
- Write a statement of work (or SOW, one of your other acronyms).
- Assign responsibilities and clarify roles.
- Track the progress of a project.
- Identify risk.

How to Create a WBS: The High-Level View

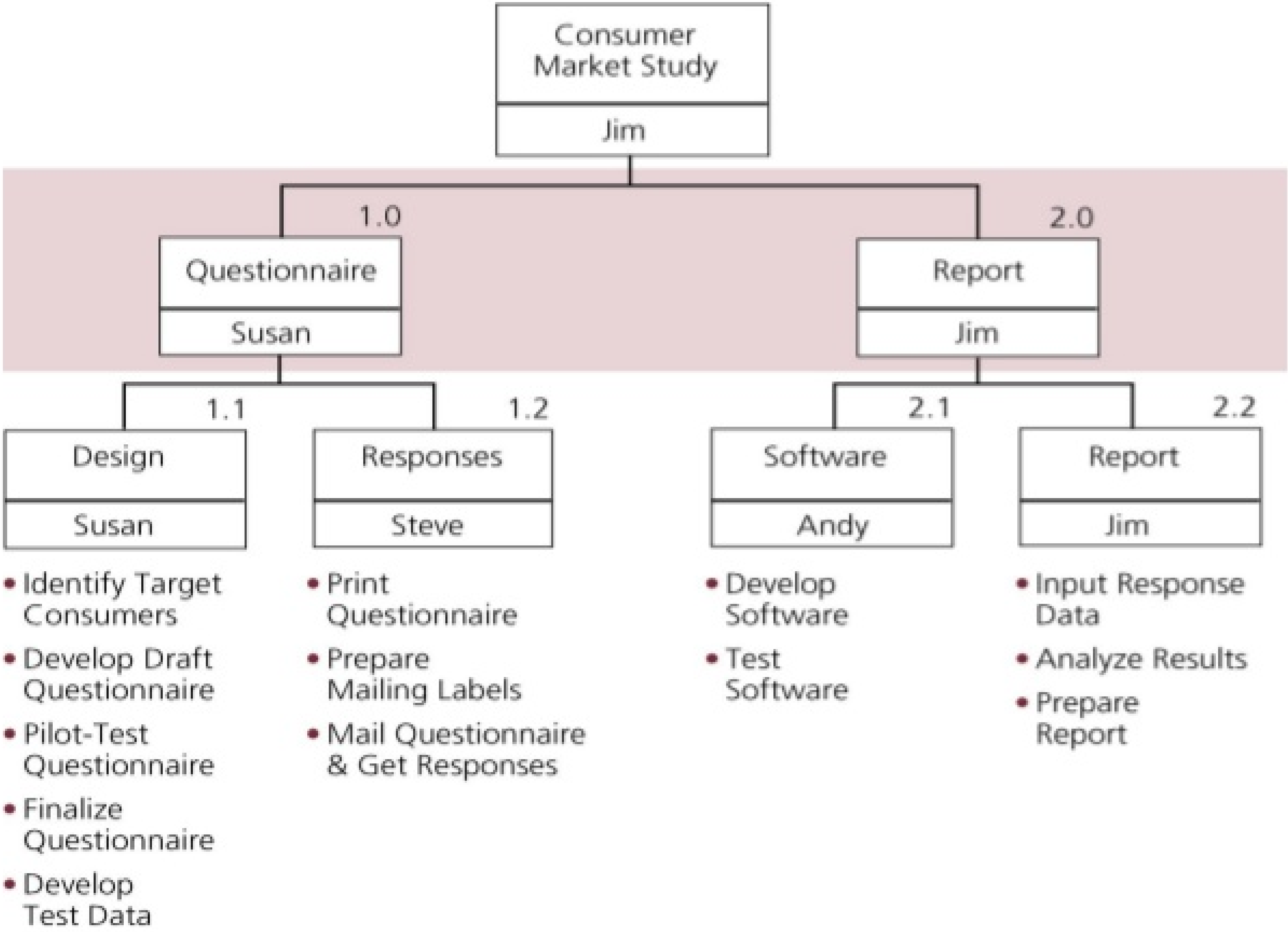
- Determine and describe the project statement.

- Highlight all the necessary phases of the project.

- Create and list the deliverables (as well as how success will be measured)

- Divide the deliverables into manageable tasks.

Work Breakdown Structure for Consumer Market Study Project



Consumer Market Study

Jim

1.0 Questionnaire

Susan

1.1 Design

Susan

- Identify Target Consumers
- Develop Draft Questionnaire
- Pilot-Test Questionnaire
- Finalize Questionnaire
- Develop Test Data

1.2 Responses

Steve

- Print Questionnaire
- Prepare Mailing Labels
- Mail Questionnaire & Get Responses

2.0 Report

Jim

2.1 Software

Andy

- Develop Software
- Test Software

2.2 Report

Jim

- Input Response Data
- Analyze Results
- Prepare Report

Planning Guidelines

- Two simple planning guidelines should be considered when a project plan is being initiated or assessed.

FIRST-LEVEL WBS ELEMENT	DEFAULT BUDGET
Management	10%
Environment	10%
Requirements	10%
Design	15%
Implementation	25%
Assessment	25%
Deployment	5%
Total	100%

The first guideline prescribes a default allocation of costs among the first-level WBS elements

DOMAIN	INCEPTION	ELABORATION	CONSTRUCTION	TRANSITION
Effort	5%	20%	65%	10%
Schedule	10%	30%	50%	10%

The second guideline prescribes the allocation of effort and schedule across the life-cycle phases

- So, WBS will be help to our project's efficiency and effectiveness, how do we go about it? First, let's look at what all we need to get started.
- There are several inputs you will need to get you off on the right foot:
  - The Project Scope Statement
  - The Project Scope Management Plan
  - Organizational Process Assets
  - Approved Change Requests - (PMBOK Guide)

PMBOK stands for Project Management Body of Knowledge and it is the entire collection of processes, best practices, terminologies, and guidelines that are accepted as standards within the project management industry.

- These above four inputs should give you all the information you and your team needs to create your WBS. Along with these inputs, you will use certain tools as well:
  - Work Breakdown Structure Templates
  - Decomposition - (PMBOK Guide)
- Finally, using these inputs and tools you will create the following outputs:
- Work Breakdown Structure
- WBS Dictionary
- Scope Baseline
- Project Scope Statement (updates)
- Project Scope Management Plan (updates)
- Requested Changes - (PMBOK Guide)

The Cost and Schedule Estimating Process

- Project plans need to be derived from two perspectives.

  - **Forward-looking:**

1. The software project manager develops a characterization of the overall size, process, environment, people, and quality required for the project.

2. A macro-level estimate of the total effort and schedule is developed using a software cost estimation model.

3. The software project manager partitions the estimate for the effort into a top-level WBS, also partitions the schedule into major milestone dates and partitions the effort into a staffing profile

4. At this point, subproject managers are given the responsibility for decomposing each of the WBS elements into lower levels using their top-level allocation, staffing profile, and major milestone dates as constraints.

- ***Backward-looking:***

1. The lowest level WBS elements are elaborated into detailed tasks, for which budgets and schedules are estimated by the responsible WBS element manager.

2. Estimates are combined and integrated into higher level budgets and milestones.

3. Comparisons are made with the top-down budgets and schedule milestones. Gross differences are assessed and adjustments are made in order to converge on agreement between the top-down and the bottom-up estimates.

Iterative Process Planning (The Iteration Planning Process)

Engineering Stage		Production Stage	
Inception	Elaboration	Construction	Transition
Feasibility iterations	Architecture iterations	Usable iterations	Product releases

Engineering stage planning emphasis:

- Macro-level task estimation for production-stage artifacts
- Micro-level task estimation for engineering artifacts
- Stakeholder concurrence
- Analysis of actual vs. planned expenditures
- Tuning the top-downconcurrence
- project-independent planning guidelines into project-specific planning guidelines.

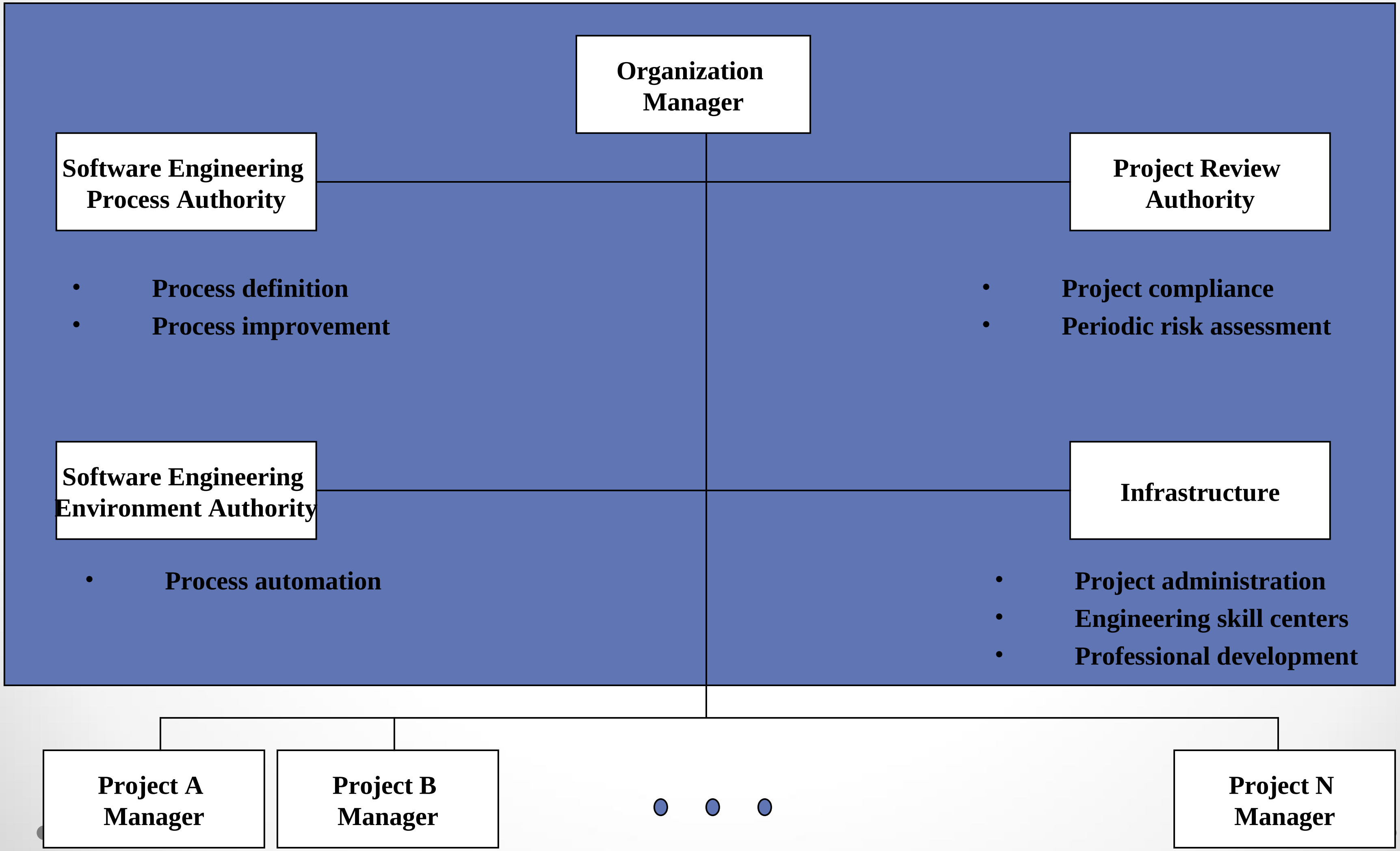
Production stage Planning emphasis:

- Micro-level task estimation for production-stage artifacts
- Macro-level task estimation for engineering artifacts
- Stakeholder concurrence
- Fine-grained variance analysis of actual vs. planned expenditures

Project Organizations and Responsibilities

[Line-of-Business Organizations]

Default roles in a software line-of-business organizations



Organization Manager

Software Engineering Process Authority

- Process definition
- Process improvement

Project Review Authority

- Project compliance
- Periodic risk assessment

Software Engineering Environment Authority

- Process automation

Infrastructure

- Project administration
- Engineering skill centers
- Professional development

Project A Manager

Project B Manager

Project N Manager

Project Organizations and Responsibilities

Project Organizations

Software Management Team

- Systems Engineering
- Financial Administration
- Quality Assurance

Artifacts

- Business case
- Vision
- Software development plan
- Work breakdown structure
- Status assessments
- Requirements set

Responsibilities

- Resource commitments
- Personnel assignments
- Plans, priorities,
- Stakeholder satisfaction
- Scope definition
- Risk management
- Project control

Life-Cycle Focus

Inception	Elaboration	Construction	Transition
Elaboration phase planning Team formulating Contract base lining Architecture costs	Construction phase planning Full staff recruitment Risk resolution Product acceptance criteria Construction costs	Transition phase planning Construction plan optimization Risk management	Customer satisfaction Contract closure Sales support Next-generation planning

Software Architecture Team:

- "A **software architect** is a **software** expert who makes high-level design choices and dictates technical standards, including **software** coding standards, tools, and platforms.”
- The most common understanding is that an **architect** is a person who takes action and dictates decisions.
- Designing, **planning** and developing are integral tasks in an architect's daily routine

Software Architecture Team

- Demonstrations
- Use-case modelers
- Design modelers
- Performance analysts

Artifacts

- Architecture description
- Requirements set
- Design set
- Release specifications

Responsibilities

- Requirements trade-offs
- Design trade-offs
- Component selection
- Initial integration
- Technical risk solution

Life-Cycle Focus

Inception	Elaboration	Construction	Transition
• Architecture prototyping • Make/buy trade-offs • Primary scenario definition • Architecture evaluation criteria definition	• Architecture base lining • Primary scenario demonstration • Make/buy trade-offs base lining	• Architecture maintenance • Multiple-component issue resolution • Performance tuning • Quality improvements	• Architecture maintenance • Multiple-component issue resolution • Performance tuning • Quality improvements

Component Team:

- A **team** that focuses on the creation of one or more **components** of a larger product that a customer would purchase.
- **Team** that is cross-functional (multi-disciplinary), single **component** focused. Contrast with feature **team**.
- **Component teams :** to build, deploy, and ultimately release.
- CT is capable of delivering end-to-end user value
- Each **team** has all the skills necessary to deliver a feature.

Software Development Team

- Component teams

Artifacts

- Design set
- Implementation set
- Deployment set

Responsibilities

- Component design
- Component implementation
- Component stand-alone test
- Component maintenance
- Component documentation

Life-Cycle Focus

Inception	Elaboration	Construction	Transition
• Prototyping support • Make/buy trade-offs	• Critical component design • Critical component implementation and test • Critical component base line	• Component design • Component implementation Component stand-alone test • Component maintenance	• Component maintenance • Component documentation

Software Assessment Team:

- *Software* process *assessments* are performed in an open and collaborative environment.
- They are for the use of the organization to improve its *software* processes, and the results are confidential to the organization.
- The organization being assessed must have members on the *assessment team*.

Software Assessment Team

- Release testing
- Change management
- Deployment
- Environment support

Artifacts

- Deployment set
- SCO database
- User manual
- Environment
- Release specifications
- Release descriptions
- Deployment documents

Responsibilities

- Project infrastructure
- Independent testing
- Requirements verification
- Metrics analysis
- Configuration control
- Change management
- User deployment

Life-Cycle Focus

Inception	Elaboration	Construction	Transition
• Infrastructure planning • Primary scenario prototyping	• Infrastructure base lining • Architecture release testing Change management • Initial user manual	• Infrastructure upgrades • Release testing • Change management • User manual base line • Requirements verification	• Infrastructure maintenance • Release base lining • Change management • Deployment to users • Requirements verification

Process Automation

[Computer-aided software engineering]

- Computer-aided software engineering (CASE) is software to support software development and evolution processes.
- Activity automation
  - o Graphical editors for system model development; Data dictionary to manage design entities;
  - o Graphical UI builder for user interface construction;
  - o Debuggers to support program fault finding;
  - o Automated translators to generate new versions of a program.

Computer-aided software engineering (CASE) Technology

- Case technology has led to significant improvements in the software process. However, these are not the order of magnitude improvements that were once predicted
  - Software engineering requires creative thought - this is not readily automated;
  - Software engineering is a team activity and, for large projects, much time is spent in team interactions. CASE technology does not really support these.

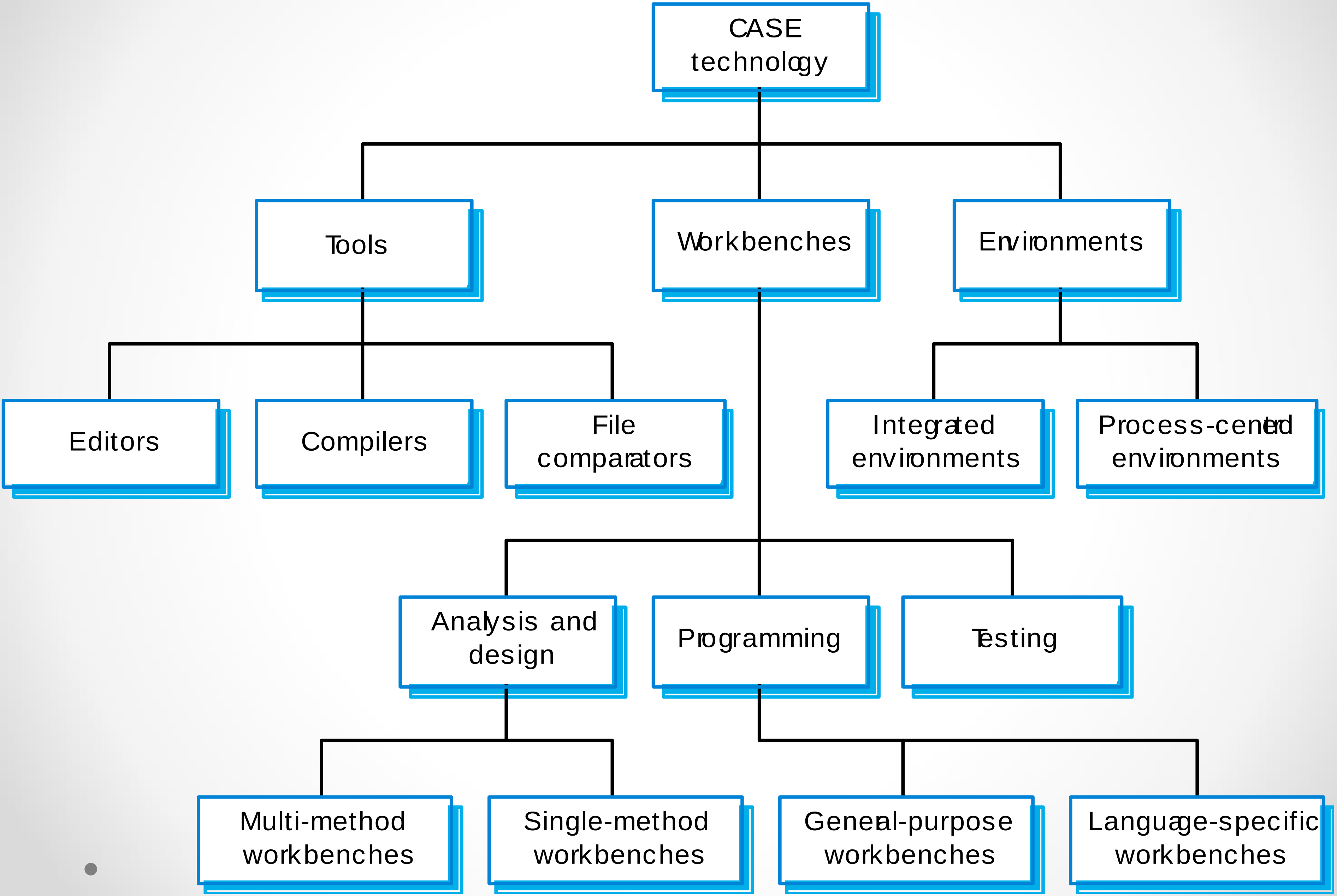
CASE Classification

- Classification helps us understand the different types of CASE tools and their support for process activities.
- Functional perspective
  - Tools are classified according to their specific function.
- Process perspective
  - Tools are classified according to process activities that are supported.
- Integration perspective
  - Tools are classified according to their organisation into integrated units.

CASE Integration

- Tools
  - Support individual process tasks such as design consistency checking, text editing, etc.
- Workbenches
  - Support a process phase such as specification or design, Normally include a number of integrated tools.
- Environments
  - Support all or a substantial part of an entire software process. Normally include several integrated workbenches.

Tools, Workbenches, Environments



Project Control and Process Instrumentation

The Core Metrics

METRIC	PURPOSE	PERSPECTIVES
Work and progress	Iteration planning, plan vs. actuals, management indicator	Source Lines Of Code, function points, object points, scenarios, test cases, Supply Chain Operating System
Budget cost and expenditures	Financial insight, plan vs. actuals, management indicator	Cost per month, full-time staff per month, percentage of budget expended
Staffing and team dynamics	Resource plan vs. actuals, hiring rate, attrition rate	People per month added, people per month leaving
Change traffic and stability	Iteration planning, management indicator of schedule convergence	Software changes
Breakage and modularity	Convergence, software scrap, quality indicator	Reworked SLOC per change, by type, by release/ component/subsystem
Rework and adoptability	Convergence, software rework, quality indicator	Average hours per change, by type, by release/ component/subsystem

Process Customization

- It is important to have visible milestones in the life cycle , where various stakeholders meet to discuss progress and planes.
  - **The purpose of this events is to:**

Process Customization

- It is important to have visible milestones in the life cycle , where various stakeholders meet to discuss progress and planes.

  - **The purpose of this events is to:**

- Synchronize stakeholder expectations and achieve concurrence on the requirements, the design, and the plan.
- Synchronize related artifacts into a consistent and balanced state
- Identify the important risks, issues, and out-of-rolerance conditions
- Perform a global assessment for the whole life-cycle.

Thank you