UNIT - VI

Project Organizations and Responsibilities: Line-of-Business Organizations, Project Organizations, evolution of Organizations.

Process Automation: Automation Building blocks, The Project Environment.

Project Organizations and Responsibilities:

- **Organizations** engaged in software Line-of-Business need to support projects with the infrastructure necessary to use a common process.
- **Project** organizations need to allocate artifacts & responsibilities across project team to ensure a balance of global (architecture) & local (component) concerns.
- **The organization** must evolve with the WBS & Life cycle concerns.
- **Software lines of business & product teams have different motivation.**
- **Software lines of business** are motivated by return of investment (ROI), new business discriminators, market diversification&profitability.
- **Project teams** are motivated by the cost, Schedule&quality of specific deliverables

1) Line-Of-Business Organizations:

The main features of default organization are as follows:

- Responsibility for process definition & maintenance is specific to a cohesive line of business.
- Responsibility for process automation is an organizational role & isequal in importance to the process definition role.
- Organizational role may be fulfilled by a single individual or several different teams.

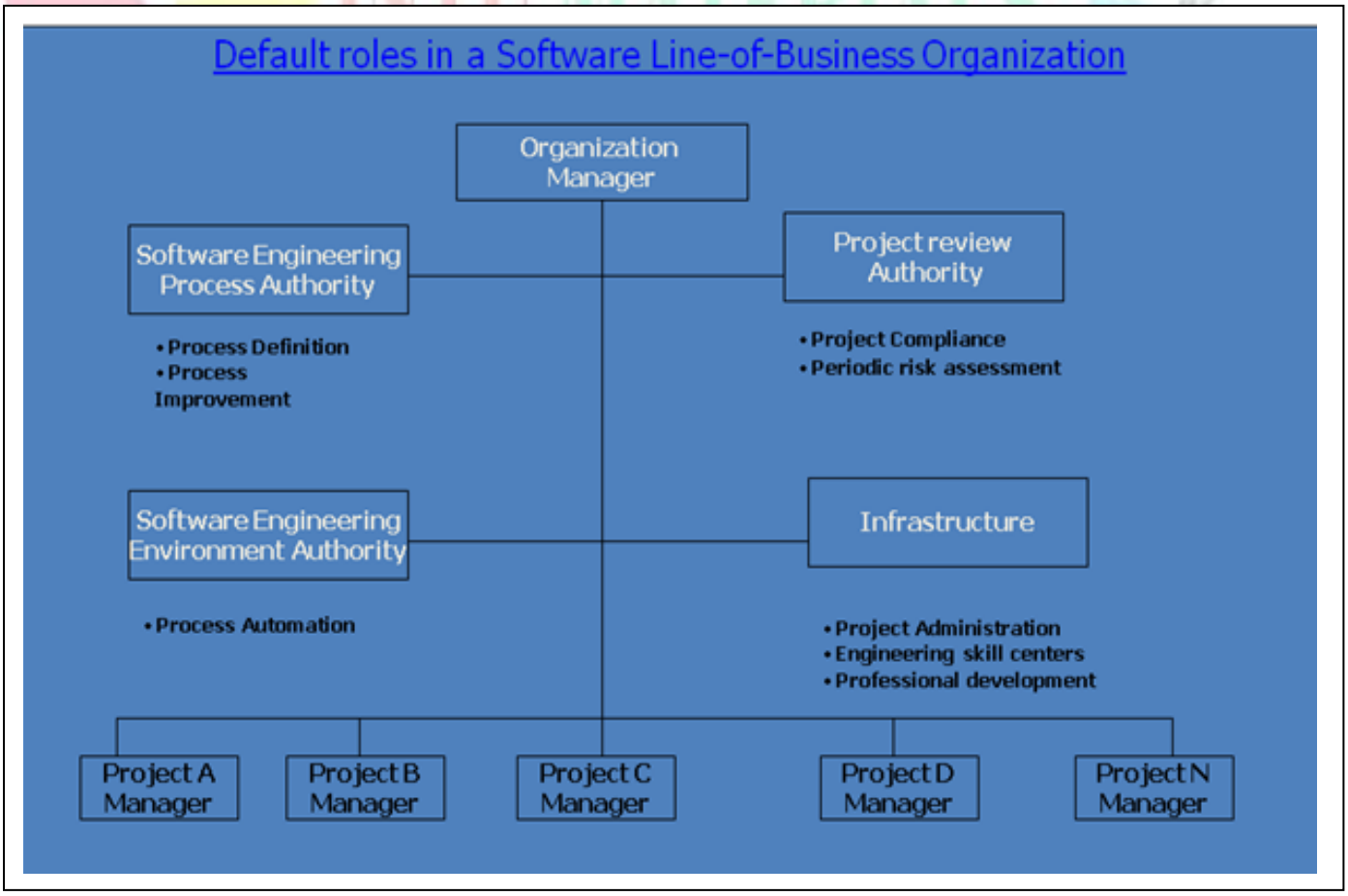


Fig: Default roles in a software Line-of-Business Organization.

Software Engineering Process Authority (SEPA)
The SEPA facilities the exchange of information & process guidance both to & from project practitioners

This role is accountable to General Manager for maintaining a current assessment of the organization's process maturity & its plan for future improvement

Project Review Authority (PRA)
The PRA is the single individual responsible for ensuring that a software project complies with all organizational & business unit software policies, practices & standards

A software Project Manager is responsible for meeting the requirements of a contract or some other project compliance standard

Software Engineering Environment Authority(SEEA)
The SEEA is responsible for automating the organization's process, maintainingthe organization's standard environment, Training projects touse the environment&maintaining organization-wide reusable assets
The SEEA role is necessary to achieve a significant ROI for common process.

Infrastructure
An organization's infrastructure provides human resources support, project-independent research & development, &other capital software engineering assets.

2) Project organizations:

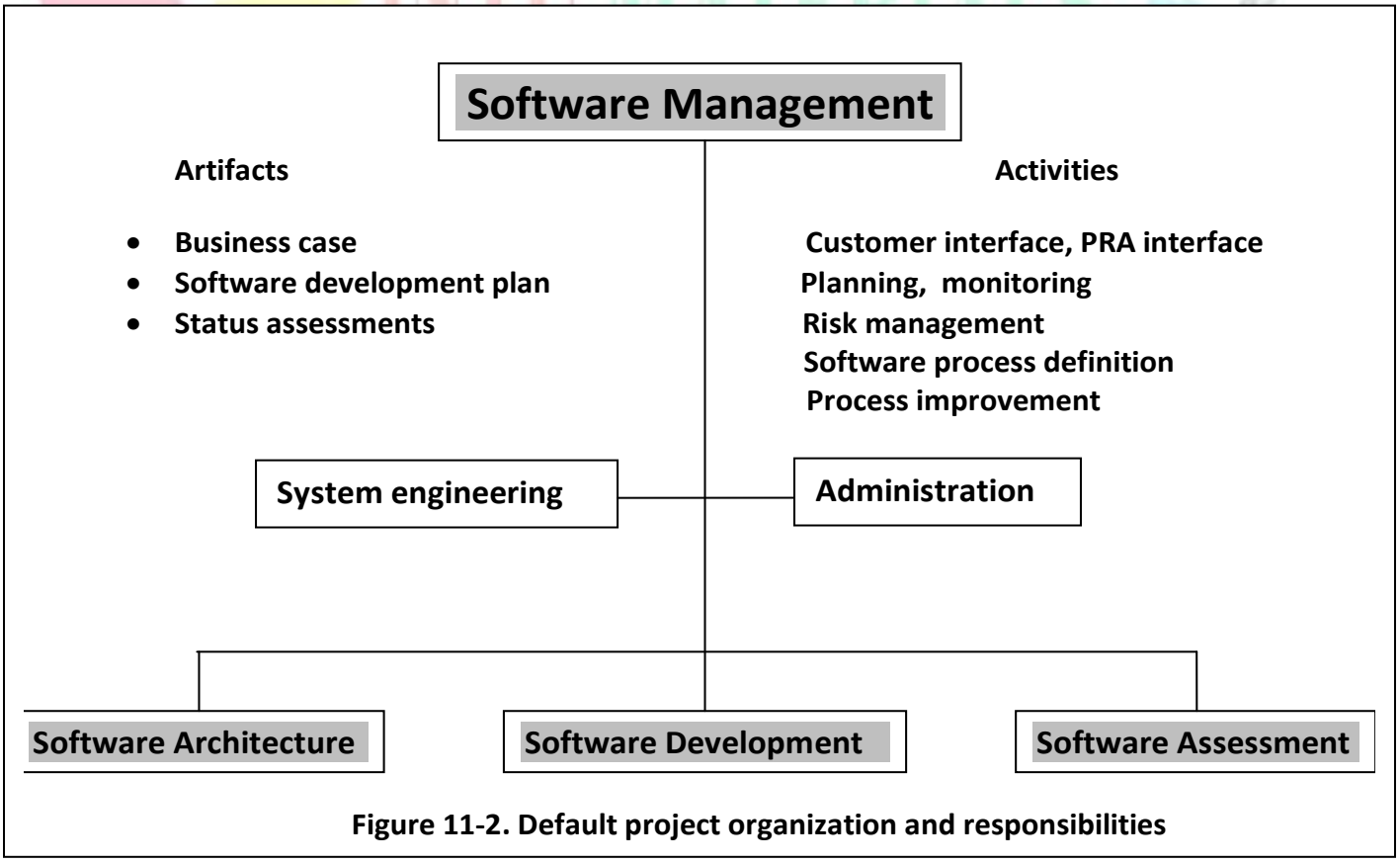
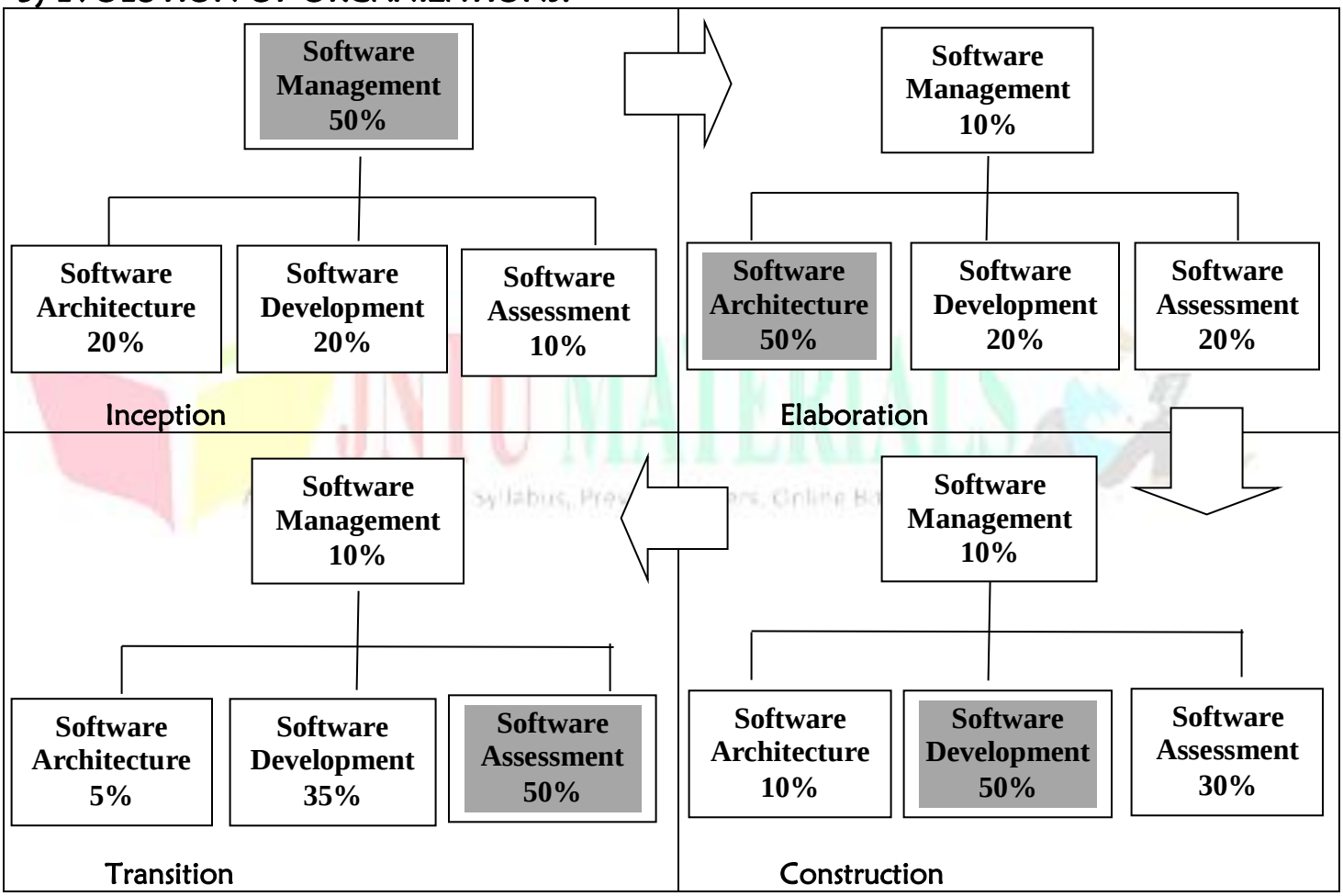


Figure 11-2. Default project organization and responsibilities

- The above figure shows a default project organization and maps project-level roles and responsibilities.
- The main features of the default organization are as follows:
- **The project management team** is an active participant, responsible for producing as well as managing.
- **The architecture team** is responsible for real artifacts and for the integration of components, not just for staff functions.
- **The development team** owns the component construction and maintenance activities.
- The assessment team is separate from development.
- **Quality** is everyone's into all activities and checkpoints.
- Each team takes responsibility for a different quality perspective.

3) EVOLUTION OF ORGANIZATIONS:



Inception: Software management: **50%** Software Architecture: 20% Software development: 20% Software Assessment (measurement/evaluation):10%	**Elaboration:** Software management: 10% Software Architecture: **50%** Software development: 20% Software Assessment (measurement/evaluation):20%
Construction: Software management: 10% Software Architecture: 10%	**Transition:** Software management: 10% Software Architecture: 5%

Software development: **50%** Software Assessment (measurement/evaluation):30%	Software development: 35% Software Assessment (measurement/evaluation):**50%**

The Process Automation:

Introductory Remarks:

The environment must be the first-class artifact of the process.

Process automation& change management is critical to an iterative process. If the change is expensive then the development organization will resist it.

Round-trip engineering& integrated environments promote change freedom & effective evolution of technical artifacts.

Metric automation is crucial to effective project control.

External stakeholders need access to environment resources to improve interaction with the development team & add value to the process.

The three levels of process which requires a certain degree of process automation for the corresponding process to be carried out efficiently.

Metaprocess (Line of business): The automation support for this level is called an infrastructure.

Macroproces (project): The automation support for a project's process is called an environment.

Microprocess (iteration): The automation support for generating artifacts is generally called a tool.

Tools: Automation Building blocks:

Many tools are available to automate the software development process. Most of the core software development tools map closely to one of the process workflows

Workflows	Environment Tools & process Automation
Management	Workflow automation, Metrics automation
Environment	Change Management, Document Automation
Requirements	Requirement Management
Design	Visual Modeling
Implementation	-Editors, Compilers, Debugger, Linker, Runtime
Assessment	-Test automation, defect Tracking
Deployment	defect Tracking

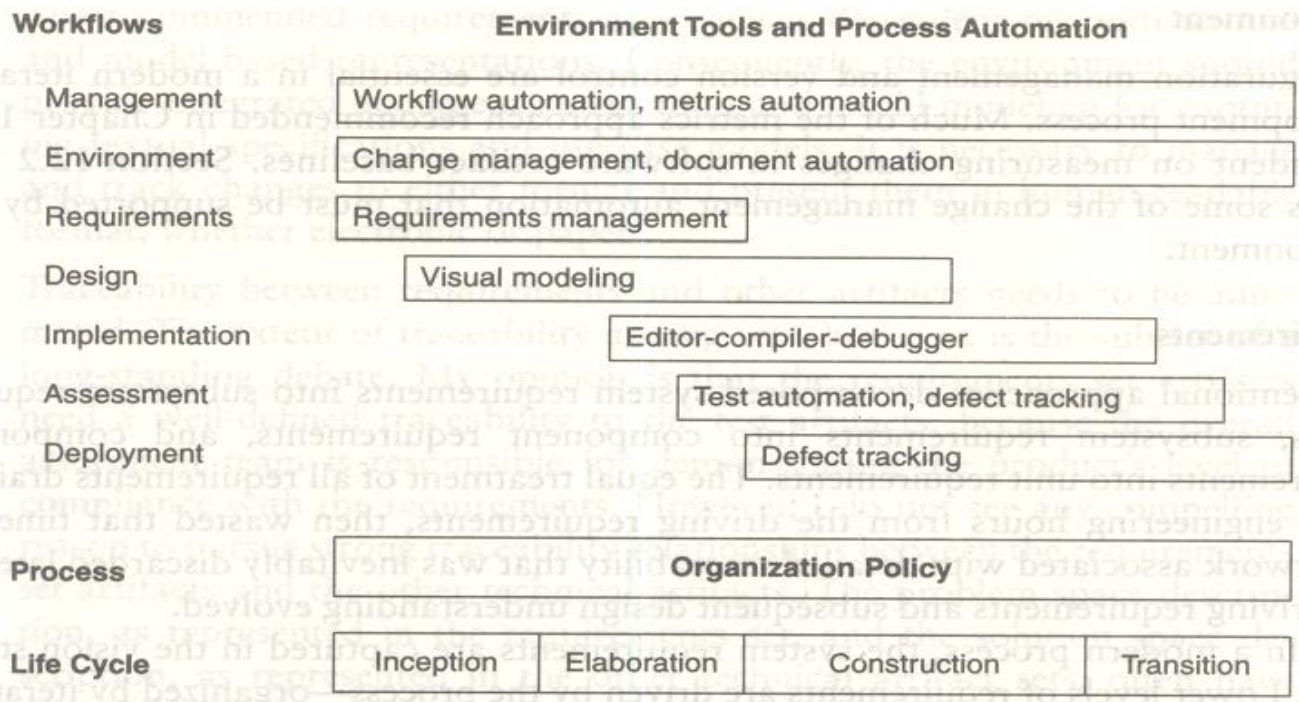


FIGURE 12-1. *Typical automation and tool components that support the process workflows*

The Project Environment:

The project environment artifacts evolve through three discrete states.
(1)Prototyping Environment.(2)Development Environment.(3)Maintenance Environment.
The **Prototype Environment** includes an architecture test bed for prototyping project architecture to evaluate trade-offs during inception & elaboration phase of the life cycle.
The **Development environment** should include a full suite of development tools needed to support variou Process workflows & round-trip engineering to the maximum extent possible.
The Maintenance Environment should typically coincide with the mature version of the development.
There are four important environment disciplines that are critical to management context & the success of a modern iterative development process.

Round-Trip engineering
Change Management
Software Change Orders (SCO)
Configuration baseline Configuration Control Board
Infrastructure
Organization Policy
Organization Environment
Stakeholder Environment.

Round Trip Environment
Tools must be integrated to maintain consistency & traceability.
Round-Trip engineering is the term used to describe this key requirement for environment that support iterative development.

As the software industry moves into maintaining different information sets for the engineering artifacts, more automation support is needed to ensure efficient & error free transition of data from one artifacts to another.

Round-trip engineering is the environment support necessary to maintainConsistency among the engineering artifacts.

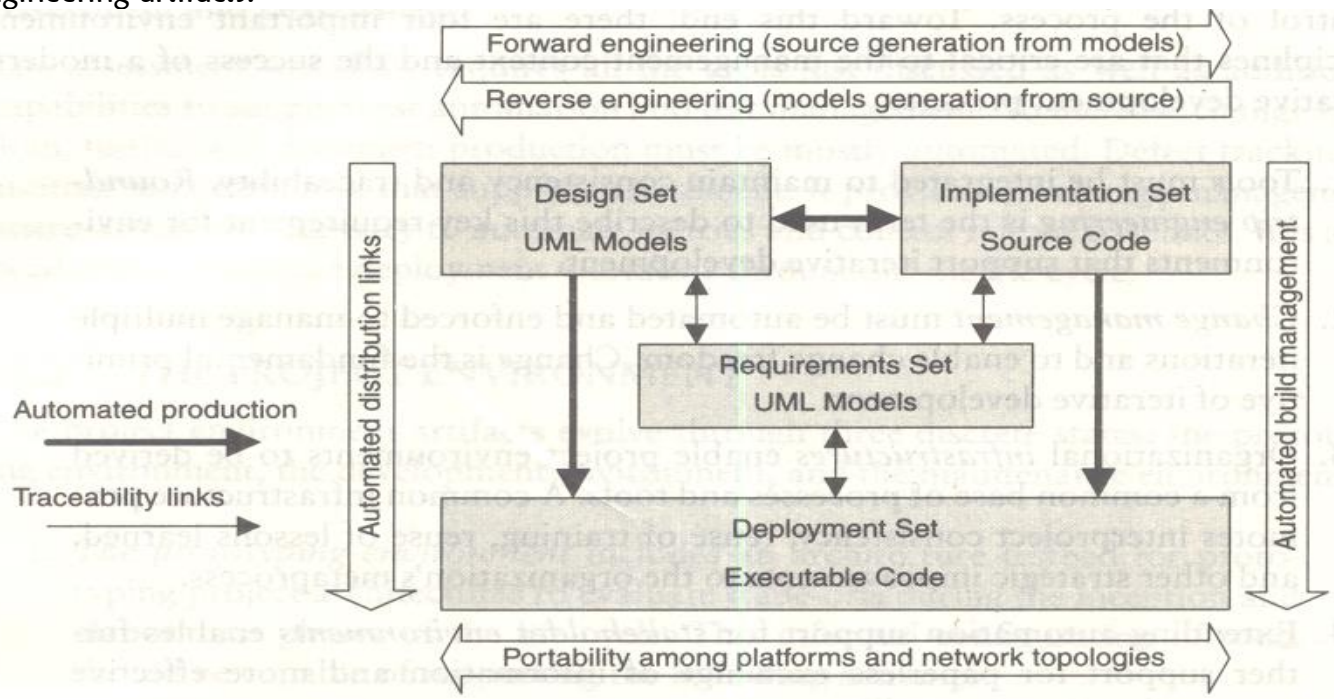


FIGURE 12-2. *Round-trip engineering*

Change Management

Change management must be automated & enforced to manage multiple iterations & to enable change freedom.

Change is the fundamental primitive of iterative Development.

I. Software Change Orders

The atomic unit of software work that is authorized to create,modify or obsolesce components within a configuration baseline is called a software change orders (SCO)

The basic fields of the SCO are Title, description, metrics, resolution, assessment & disposition

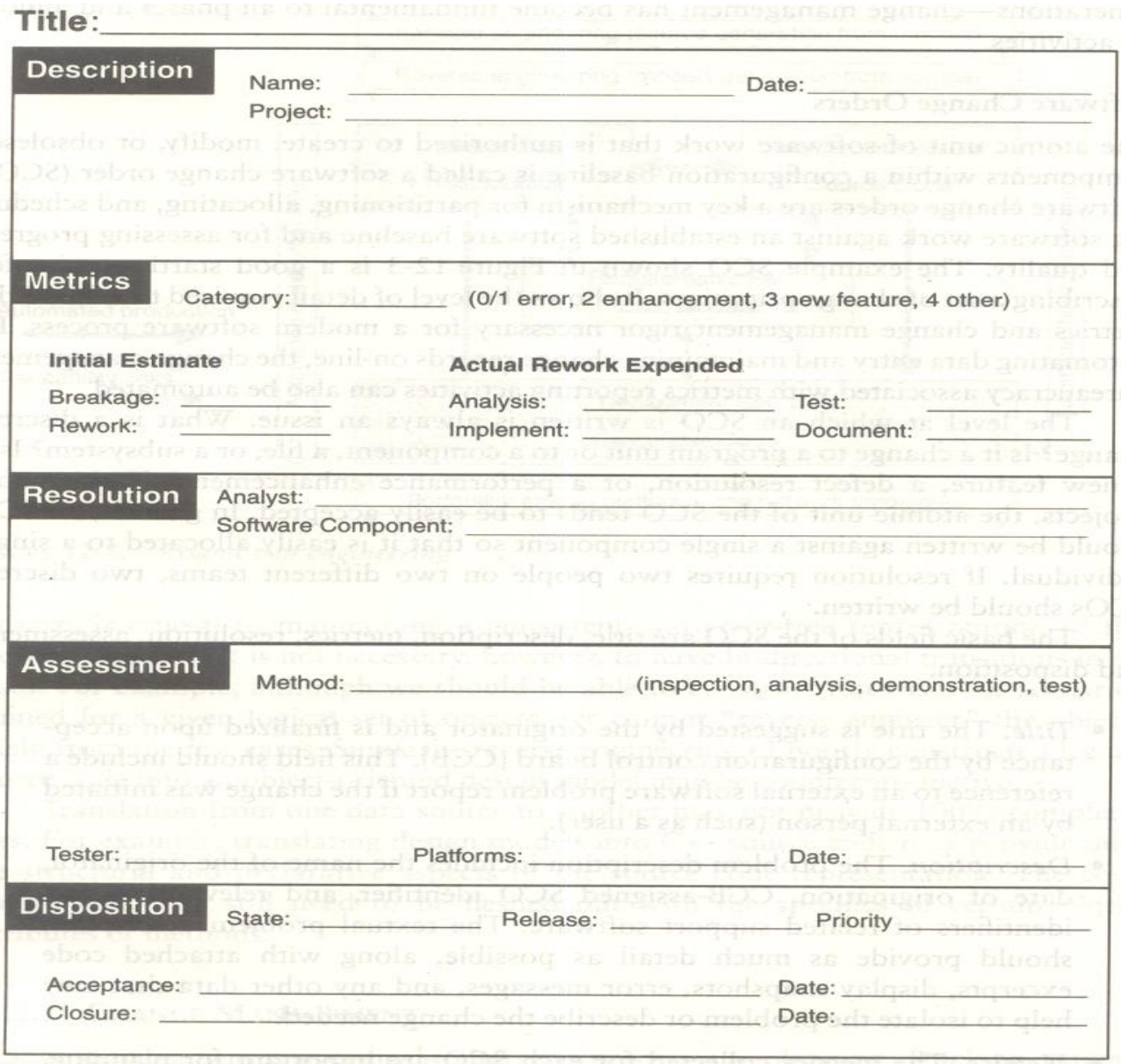


FIGURE 12-3. *The primitive components of a software change order*

Change management

II.Configuration Baseline

A configuration baseline is a named collection of software components &Supporting documentation that is subjected to change management & is upgraded, maintained, tested, statuses & obsolesced a unit

There are generally two classes of baselines

External Product Release

Internal testing Release

Three levels of baseline releases are required for most Systems

1. Major release (N)
2. Minor Release (M)
3. Interim (temporary) Release (X)

Major release represents a new generation of the product or project

A minor release represents the same basic product but with enhanced features, performance or quality.

Major & Minor releases are intended to be external product releases that are persistent & supported for a period of time.

An interim release corresponds to a developmental configuration that is intended to be transient.

Once software is placed in a controlled baseline all changes are tracked such that a distinction must be made for the cause of the change. Change categories are

Type 0: Critical Failures (must be fixed before release)

Type 1: A bug or defect either does not impair (Harm) the usefulness of the system or can be worked around

Type 2: A change that is an enhancement rather than a response to a defect

Type 3: A change that is necessitated by the update to the environment

Type 4: Changes that are not accommodated by the othercategories.

Change Management

III Configuration Control Board (CCB)

A CCB is a team of people that functions as the decision

Authority on the content of configuration baselines

A CCB includes:

1. Software managers

2. Software Architecture managers

3. Software Development managers

4. Software Assessment managers

5. Other Stakeholderswho are integral to the maintenance of the controlled software delivery system?

Infrastructure

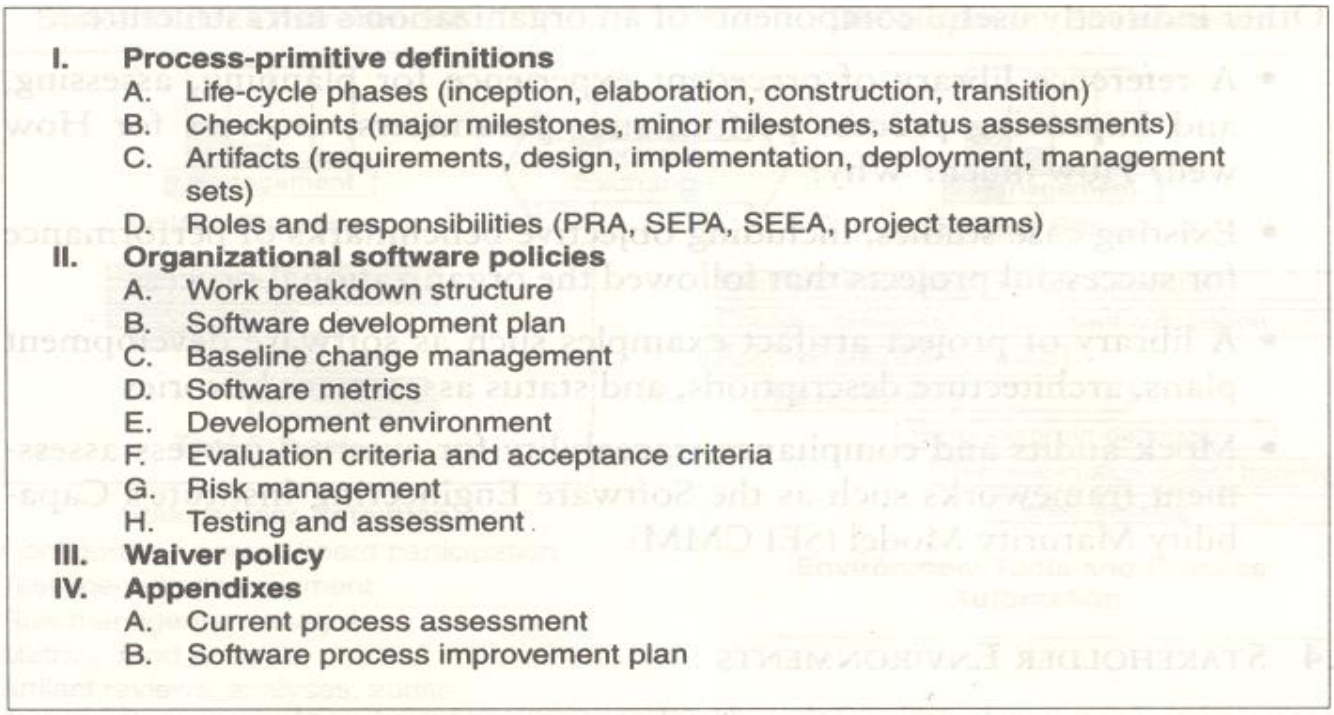
The organization infrastructure provides the organization's capitalassets including two key artifacts - Policy & Environment

I Organization Policy:

A Policy captures the standards for project software developmentprocesses

The organization policy is usually packaged as a handbook that defines the life cycles & the process primitives such as

- Major milestones
- Intermediate Artifacts
- Engineering repositories
- Metrics
- Roles & Responsibilities



I. **Process-primitive definitions**
   A. Life-cycle phases (inception, elaboration, construction, transition)
   B. Checkpoints (major milestones, minor milestones, status assessments)
   C. Artifacts (requirements, design, implementation, deployment, management sets)
   D. Roles and responsibilities (PRA, SEPA, SEEA, project teams)

II. **Organizational software policies**
   A. Work breakdown structure
   B. Software development plan
   C. Baseline change management
   D. Software metrics
   E. Development environment
   F. Evaluation criteria and acceptance criteria
   G. Risk management
   H. Testing and assessment

III. **Waiver policy**

IV. **Appendixes**
   A. Current process assessment
   B. Software process improvement plan

FIGURE 12-5. *Organization policy outline*

Infrastructure

II Organization Environment

The Environment that captures an inventory of tools which are building blocks from which project environments can be configuredefficiently & economically

Stakeholder Environment

Many large scale projects include people in external organizationsthat represent other stakeholders participating in the development processthey might include

- Procurement agency contract monitors
- End-user engineering support personnel
- Third party maintenance contractors
- Independent verification & validation contractors
- Representatives of regulatory agencies & others.

These stakeholder representatives also need to access to development resources so that they can contribute value to overall effort. These stakeholders will be access through on-line

An on-line environment accessible by the external stakeholdersallow them to participate in the process a follows

Accept & use executable increments for the hands-on evaluation.

Use the same on-line tools, data & reports that the development organization uses to manage & monitor the project

Avoid excessive travel, paper interchange delays, format translations, paper * shipping costs & other overhead cost

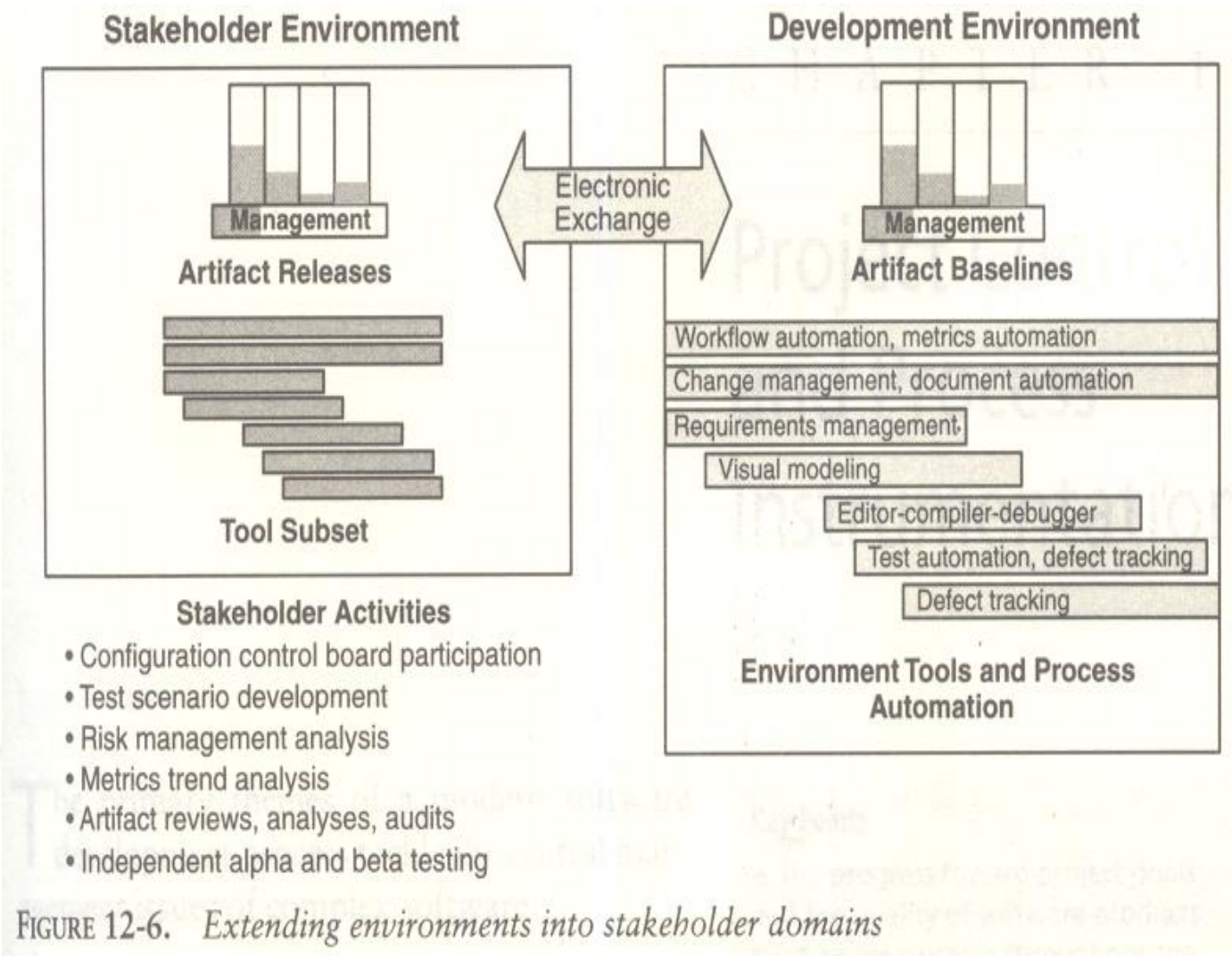


FIGURE 12-6. *Extending environments into stakeholder domains*