Process Models and Metrics

PROCESS MODELS AND METRICS

These models and metrics capture information about the processes being performed

We can model and measure the

- definition of the process
- process performers conformance to the process definition
- process performers understanding of the domain
  - to which the process is being applied

Process models and metrics can be used to

- evaluate the process
- gain insight into the product
- find weakness in the environment in which the process is being applied
- provide insight into process improvement
- help tailor and evolve the processes over time

PROCESS MODELS AND METRICS

Defining Process Terminology

Technique: A series of steps (requiring skill) for producing a desired effect, i.e., constructing or assessing software, e.g., testing, reading

Method: An organized approach for applying techniques, e.g., design inspections, test plan. Should include entry and exit criteria (when, how, how long to apply) and management supports (evaluation criteria)

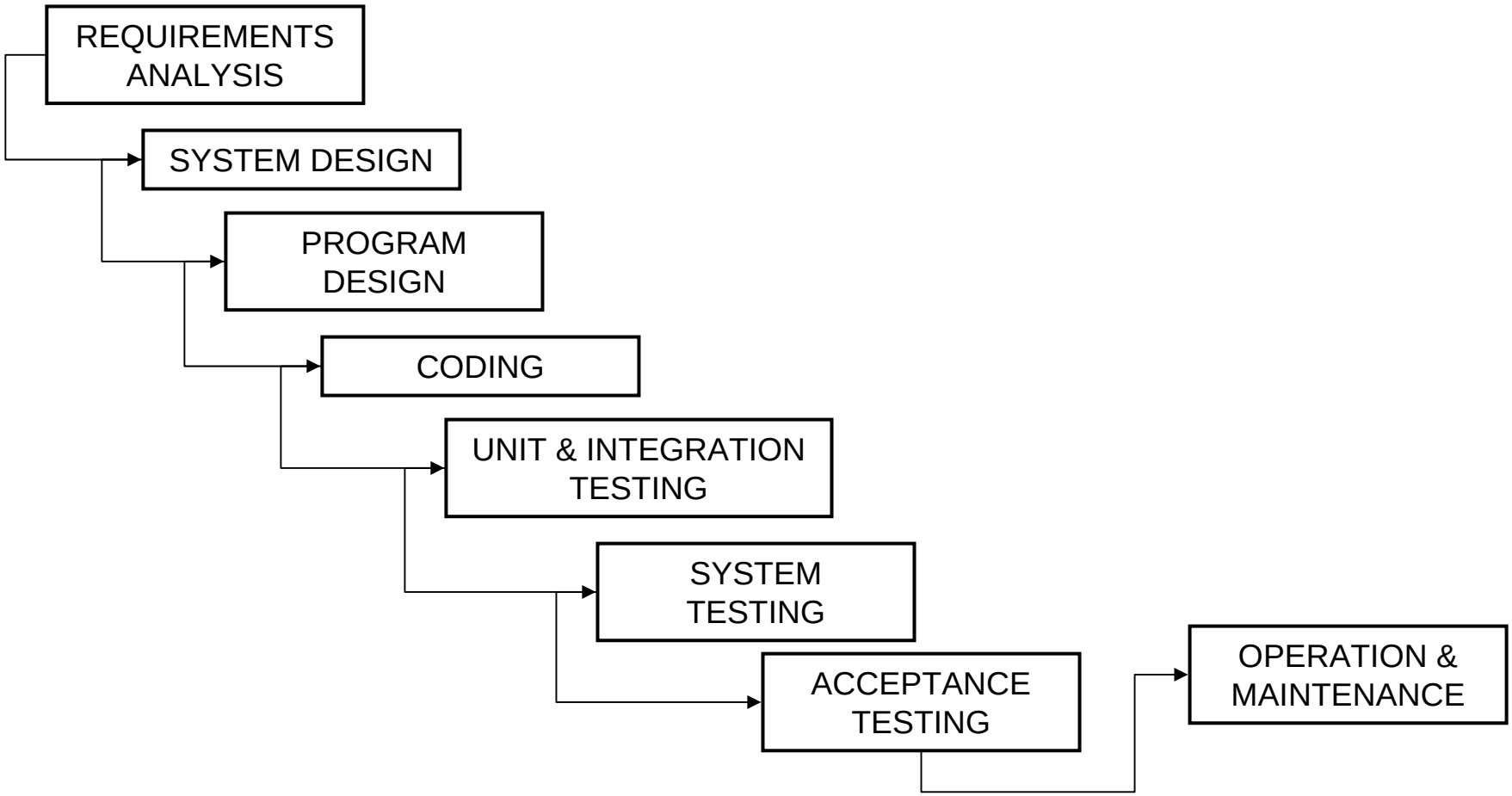
Life Cycle Model: an integrated set of methods that cover the entire life cycle, e.g., an incremental development model, using structured design, design inspections, etc.

Engineering: The application and tailoring of techniques, methods, and life cycle models to the problem, project, and organization

Life Cycle Models

Waterfall Model

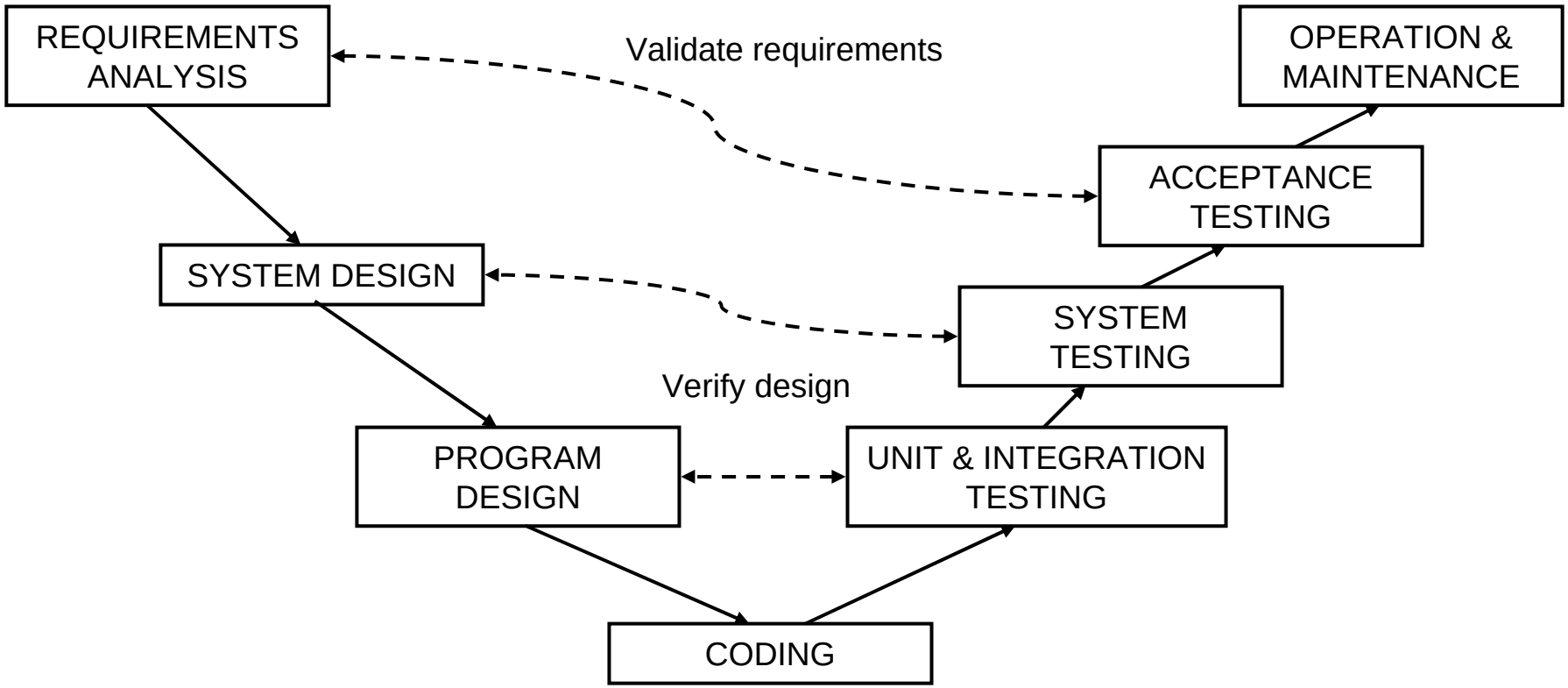
1. Start with the requirements and completely determine them
2. Pass through each of the phases sequentially
3. Loop back and update a document when appropriate



Life Cycle Models

V-Model

Same as the Waterfall model except for the emphasis on the pairing of test stages with development documents



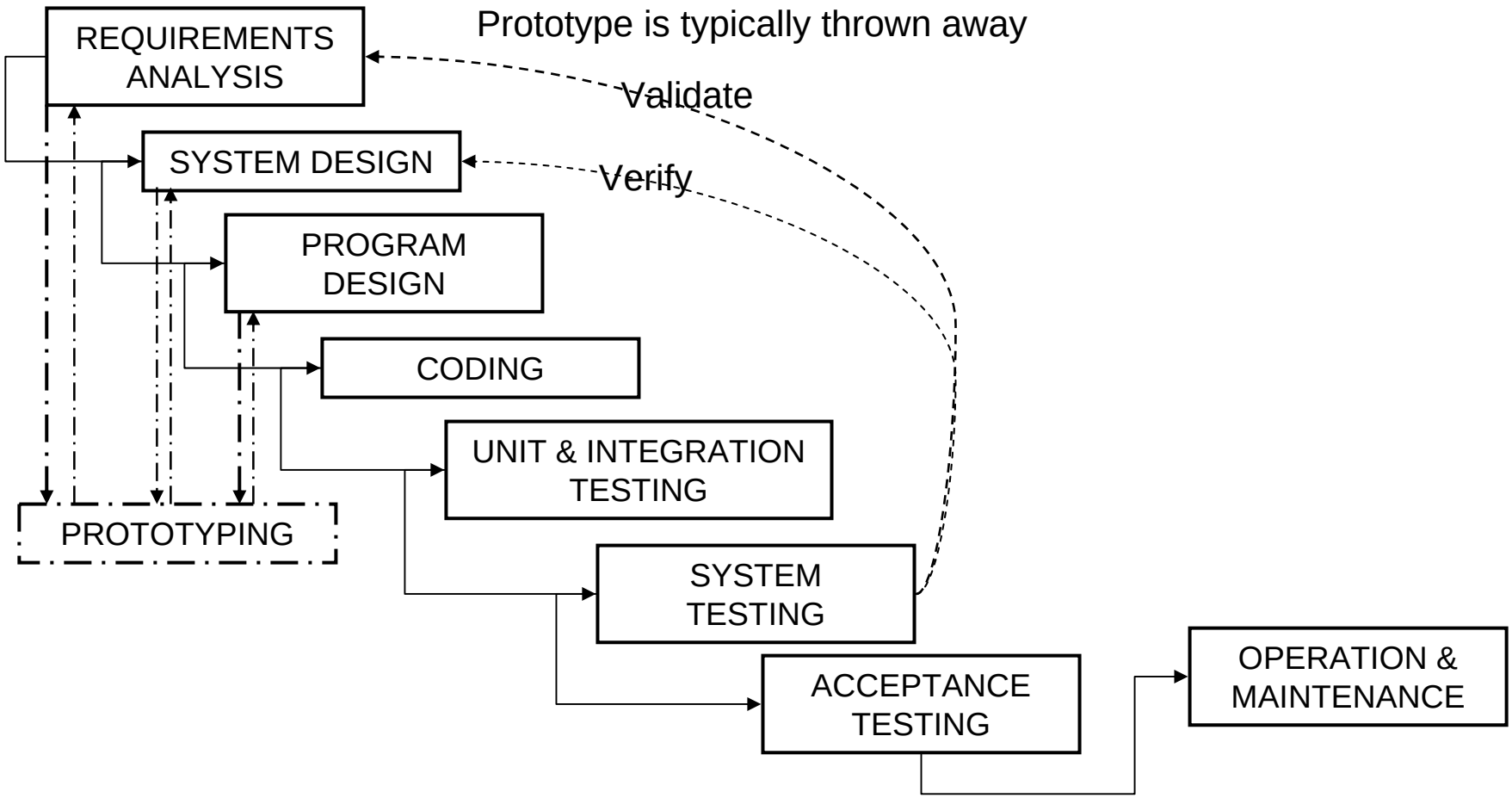
Life Cycle Models

Prototyping Model

1. Find the part of the system in need of further study
2. Build a prototype directly or build a simulation

   Prototype is typically built in a very high level language

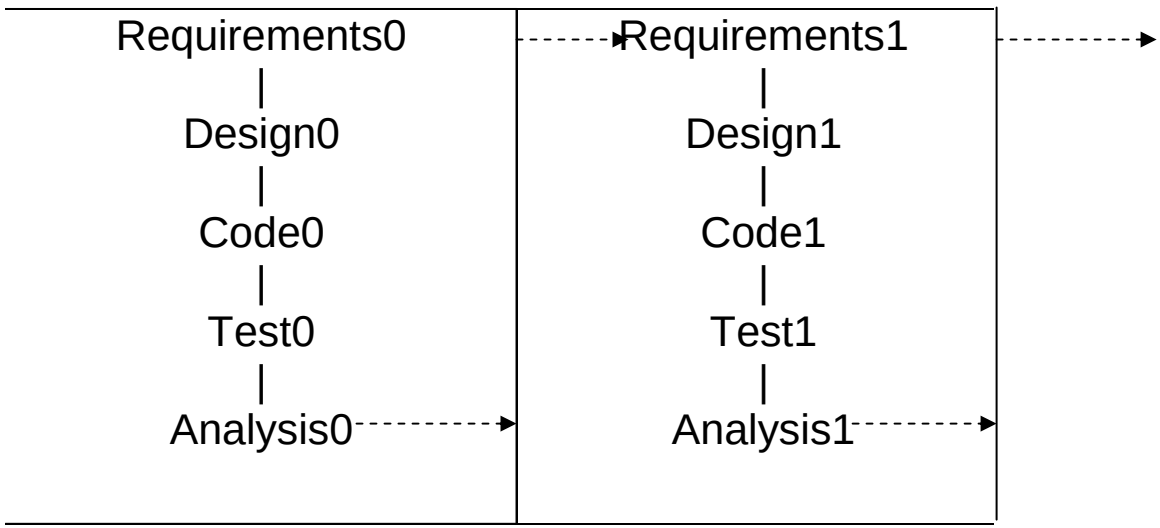
   Prototype is typically thrown away



Life Cycle Models

Iterative Enhancement Model

1. Identify the potential set of increments based upon the requirements and the learning needs.
2. Select the next subset of the requirements to be implemented
3. Design, code, test, and analyze the current increment
4. At each step of the evolutionary process, implement the next requirement set and continue to redesign, based upon analysis



Life Cycle Models

The Spiral Model

It is an iterative enhancement style, risk driven model that allows the developer to choose the life cycle model based upon the risks involved at each stage (version) of development.

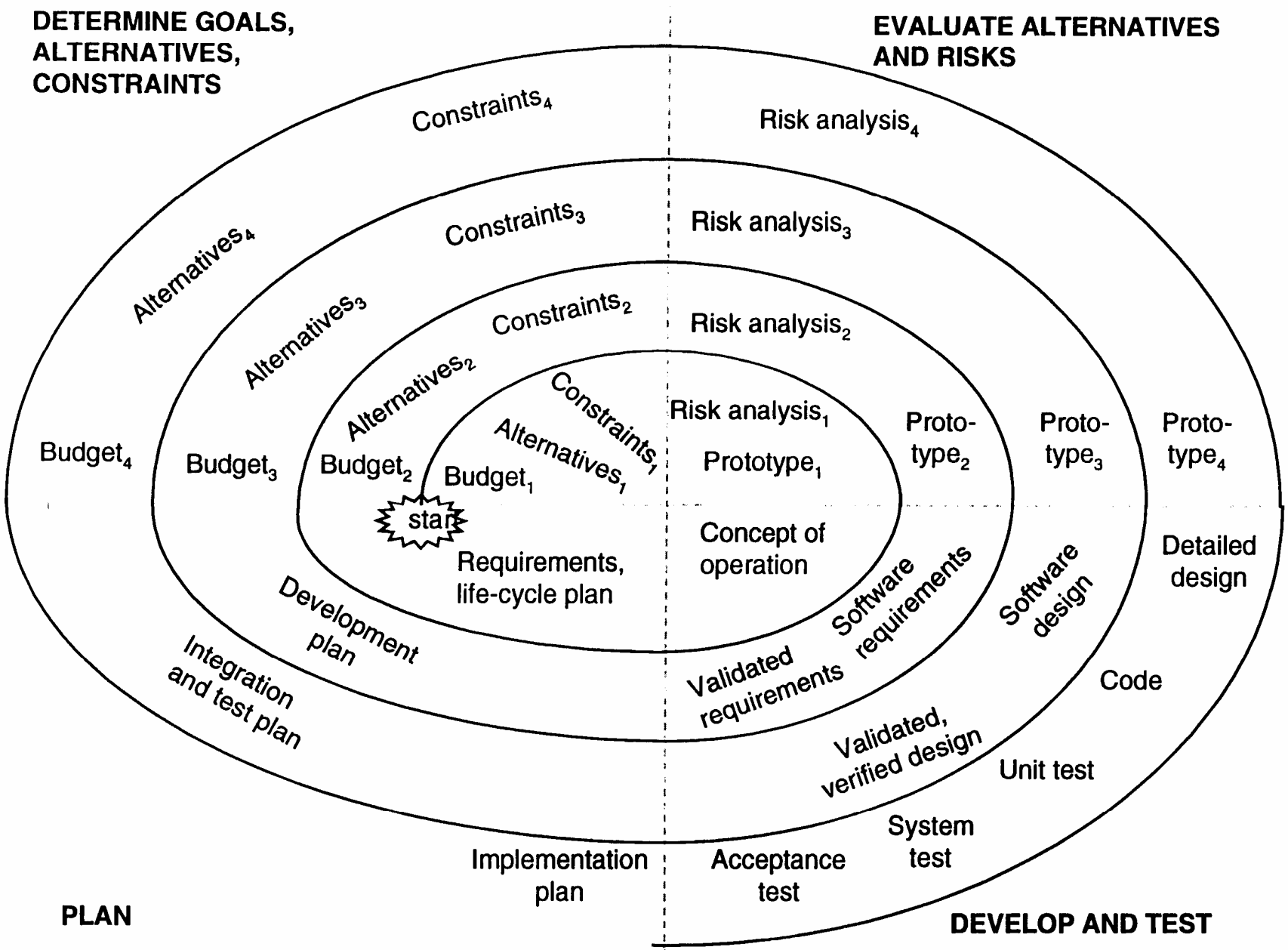
Dimensions:

- radial - cumulative cost incurred
- angular - progress made in completing each cycle of the spiral

Each cycle has the following phases:

- Identify the objectives for that cycle and the alternatives that are possible for achieving those objectives
- Evaluate the different alternatives based upon objectives and constraints (identify uncertainties and risks)
- Develop strategies that resolve the uncertainties and risks (using benchmarking, prototyping, simulation) and develop the software
- Plan the next stage, allowing any of the possible life cycle models to be used

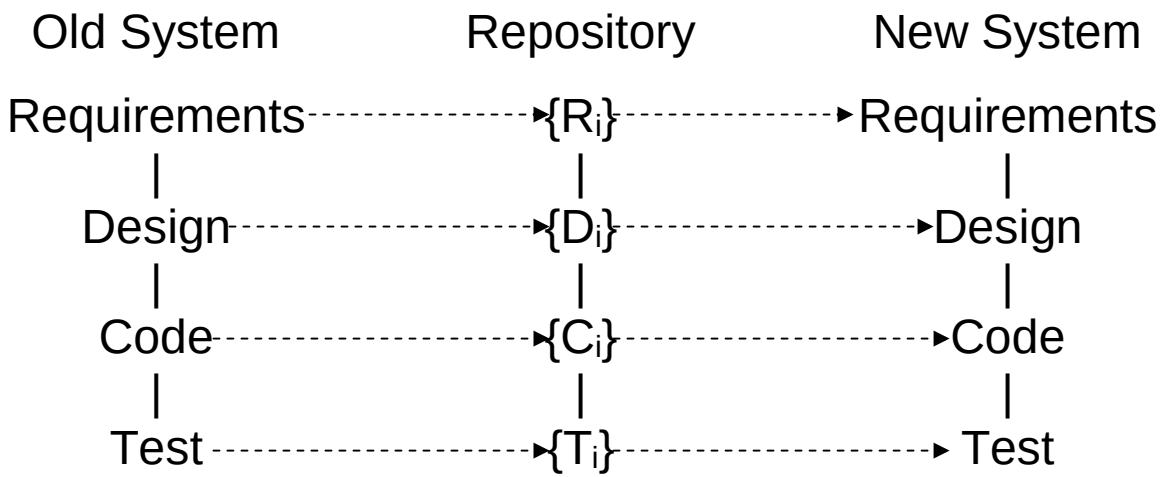
To initialize the spiral, the feasibility of the basic project objectives are analyzed and any environmental needs are addressed


DETERMINE GOALS, ALTERNATIVES, CONSTRAINTS
EVALUATE ALTERNATIVES AND RISKS
Constraints4
Risk analysis4
Alternatives4
Constraints3
Risk analysis3
Alternatives3
Constraints2
Risk analysis2
Alternatives2
Constraints1
Alternatives1
Risk analysis1
Prototype1
Budget4
Budget3
Budget2
Budget1
start
Requirements, life-cycle plan
Concept of operation
Proto-type2
Proto-type3
Proto-type4
Detailed design
Development plan
Software requirements
Software design
Integration and test plan
Validated requirements
Code
Validated, verified design
Unit test
System test
Implementation plan
Acceptance test
PLAN
DEVELOP AND TEST

Life Cycle Models

Reuse Development Model

1. State the requirements for the new system, reusing as much of the old system as feasible
2. Build a new system, using components from the old system or other systems available in the repository, developing new components where appropriate



Choosing the Life Cycle Model

Waterfall model

- Sequential
- Good when problem and solution well understood

Iterative enhancement model

- Incremental versions developed
- Good when problem or solution not well understood, schedule for full function a risk, requirements changing

Prototyping model

- Experimental version
- Good when user unsure of needs, some aspect of the system unclear, experimental analysis needed

Spiral model

- Risk oriented iterative enhancement
- Good when high risk, customer unclear of evolving requirements

Process Models And Metrics

Producer/Consumer Model

For each document or intermediate product

- who are the consumers?
- What are there operational profiles?
- What are their goals or needs for that document or product?

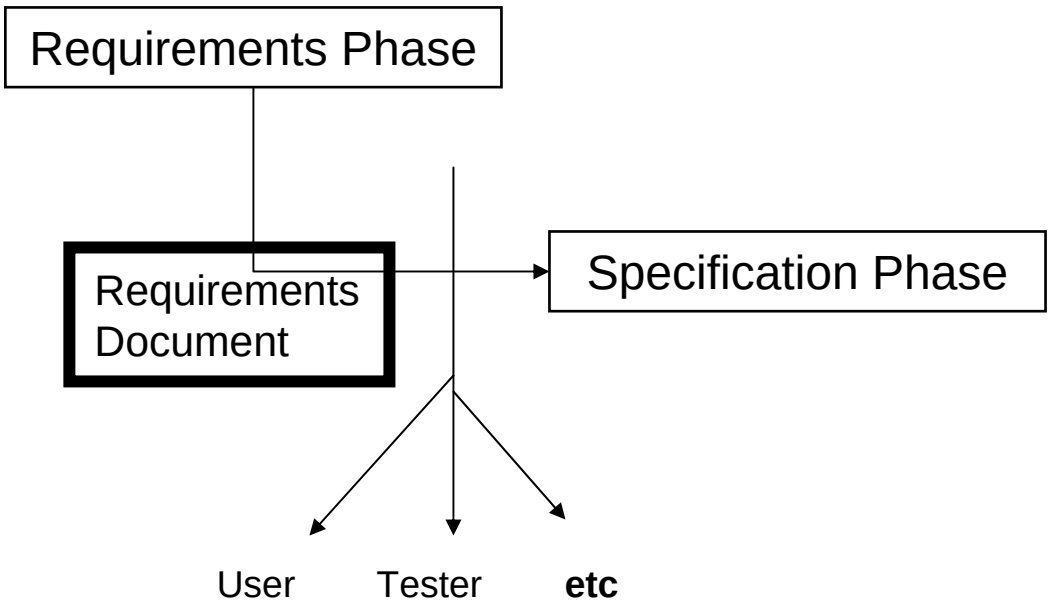
Each document should be evaluated from the perspective of all customers

Example: Requirements Document

Consumers: User, Tester, Developer, Maintainer, SQA, Hardware, …

Process Models And Metrics

Producer/Consumer Model



Each phase can be gated to allow for an evaluation from different perspectives

Process Models And Metrics

Producer/Consumer Model

Perspectives

Customer
System Analyst
Designer
Coder
System Tester
Maintainer
Quality

Process Models And Metrics

Producer/Consumer Model

	Reviewed Document				
Perspective	Requirements	Specification	Design	Code	Test
Customer	X	X			
System Analyst	X		X		
Designer		X		X	X
Coder			X		
System Tester	X			X	
Maintainer		X	X	X	
Quality					X

Choosing the Technique: Reading

Dimensions

Input object: Requirements, specification, design, code, test plan,…

Output object: Set of anomalies

Approach: Sequential, path analysis, stepwise abstraction,…

Formality: Reading, correctness demonstrations,…

Emphasis: Fault detection, traceability, performance,…

Method: Walk-throughs, inspections, reviews,…

Consumers: User, designer, tester, maintainer,…

Product qualities: Correctness, reliability, efficiency, portability,…

Process qualities: Adherence to method, integration into process,…

Quality view: Assurance, control,...

Choosing the Technique: Testing

Dimensions

Input object: System, subsystem, feature, module,…

Output object: Test results

Approach: Structural, functional, error-based, statistical testing,...

Formality: Full adherence, partial adherence,...

Emphasis: Fault detection, new features, reliability, performance,...

Method: As specified in the test plan

Consumers: Various classes of customer/hardware configurations

Product qualities: Reliability, efficiency,…

Process qualities: Adherence to method, integration into process,…

Quality view: Assurance, control,...

Choosing the Technique: Stepwise Refinement

Dimensions

Input object: Requirements/specification document, higher level design (possibly mix of old and new),...

Output object: Some level of design document

Approach: Functional/state machine decomposition, axiomatic, Jackson,...

Formality: Tree charts, PDL, functions, predicate calculus,...

Emphasis: Information hiding, strength/coupling, low complexity,...

Method: Top down design, builds, various review/exit criteria,...

Consumers: Coder, tester, maintainer,…

Product qualities: Correctness, efficiency, portability, reusability,…

Process qualities: Adherence to method, integration into process,…

Methodology Needs: e.g., Testing

Based upon project characteristics, project goals, the state of the project objects, and prior environmental history

- generate goals for testing
- decide test phases and goal of each phase
- decide techniques to be used in each phase

For each test phase determine

- process definition
- techniques to be used
- experience models to be used
- entry and exit criteria
- the match of entry and exit criteria for consecutive phases

Project characteristics make it possible to find similar prior projects for experience

Experience models used include

- prior defect and resource distributions associated with phases and techniques
- technique histories including strengths and weaknesses associated with various requirements, etc.

Defining Process Goals

What are the goals of the Software Development Process?

From the project management perspective:

Develop a set of documents that all represent the same system

From the user's perspective:

Develop a system that satisfies the user's needs with respect to functionality, quality, cost, etc.

From the corporate perspective:

Improve the organization's ability to develop quality systems productively and profitably

Defining Process Goals

Given process goals, the selection of the activities depend upon the environmental characteristics

What are the goals of the requirements activity?

To characterize the user's needs in order to specify a system that satisfies them

The selection of activities depends upon the environment, e.g., contract vs. general product

What does this say about user input?

How do we create customer/user scenarios and models?

How do we get the user involved in the requirements definition (e.g., prototype screens), and test plans?

Defining Process Goals

Goals help define the life cycle model, methods, and techniques

What are the goals of the test activity?

To assess quality or to find failures?

- Selection of activities depends upon the answer

- Assess quality:
  - Tests based upon user operational scenario
  - Statistically based testing technique
  - Reliability modeling for assessment

- Find failures:
  - Test a function at a time
  - General to specific
  - Reliability models not appropriate

Defining Process Guidebook

Need to provide flexible process definition
appropriate information for process selection
support process integration and configuration,
via tailorable definitions and characterizations for life cycle models, methods and techniques

Examples:

If problem and solution well understood
choose **waterfall process model**

If high number of faults of omission expected
emphasize **traceability reading** approach embedded in **design inspections**

When embedding **traceability reading** in **design inspections**, make sure **traceability matrix** exists

Process Needs

There is a reusable software technology, but it is not simple

We need

- flexible definitions that can be easily tailored to the problem
- goals for processes

We need experimentation and analysis of the various technologies to understand

- their strengths and weaknesses
- when and where they are appropriate
- how to tailor them to a specific project

We need education and training in the

- application
- life cycle models
- methods
- techniques
- tools

e.g. what training is done in reading?

Process Models and Metrics
Building Measurable Process Models

Modeling the education and training process

Assume the organization has a process in place for training with respect to a method or technique

In our example, let us assume a simple process consisting in a set of steps:

1. provide the individual with training manuals they must read
2. provide a course, educating the individual in the process
3. provide training by applying the process to a toy problem
4. assign the individual to a project that is using the process, mentored by an experienced method user
5. after this the individual is considered fully trained in the process

Process Models and Metrics
Building Measurable Process Models

Modeling the education and training process

If we need to capture the experience of an individual with respect to a particular method or technique, we can convert this process to an operational model by associating ordinal values with the process steps, representing various stages of experience maturity, by letting each step represents a further passage along the ordinal scale

Thus a value of

0 implies no training,

1 implies the individual has read the manuals,

2 implies the individual has been through a training course,

3 implies the individual has had experience in a laboratory environment,

4 implies the process had been used on a project before, under tutelage,

and 5 implies the process has been used on several projects

Process Models and Metrics
Building Measurable Process Models

Measuring the education and training process

If the education and training process model is valid, the ***ordinal scale ratings*** are valid

We can us this ordinal scale rating on a questionnaire

Characterize your experience with method X(subjective rating per person)

0 - none
1 - have read the manuals
2 - have had a training course
3 - have had experience in a laboratory environment
4 - have used on a project before
5 - have used on several projects before

Process Models and Metrics
Building Measurable Process Models

Measuring the education and training process

Now let us assume we want to characterize the experience of a team with respect to method X,

How do we combine individual data to create a team model?

We can build an interpretive model of the form:

- 0 implies no member of the team has more than a 1
- 1 implies the team lead has a 4 and each team member has at least a 2
- 2 implies the team lead has a 5 and each team member has at least a 3
- 3 implies the team lead has a 5 and each team member has at least a 3 and half the team has at least a 4

This model can be tested over time and can be improved with experience

Process Models and Metrics
Building Measurable Process Models

Issues

How do you define a process?

How do you define a process so that it is measurable?

What do you measure? The steps? The key characteristics? How do you weight them?

How do you know if a process has been applied? Applied correctly? How do you check process conformance?

How do you know if the developer is capable of applying it correctly?

Can measurement be used to give advice in the application? Help accentuate what is important?

REVIEW PROCESS COMPLIANCE (RPC)

Q1: Were the basic characteristics of a review process carried out sufficiently to meet process conformance syntactically? (evaluate read time, calendar time, meeting format, meeting appropriateness) (0-5)

- 0 - Nothing was done correctly
- 1 - Major steps skipped or badly performed
- 2 - Some major steps skipped or contained major flaws
- 3 - All major steps were performed to a reasonable degree, but there were minor flaws
- 4 - All major steps were performed correctly
- 5 - All steps were performed correctly

Q1.1: How much time did each reviewer take to do the reading? (in hours)

Q1.2 Was the amount of calendar time given for the review ample for each reviewer? (y/n)

Q1.3 Were the purpose and agenda followed? (0-3)

- 0 - Not at all or no agenda was given
- 1 - The agenda was heavily revised
- 2 - The agenda was modified slightly
- 3 - Agenda was followed as given

Q1.4 Was a go/no go decision to hold the meeting correctly called? (y/n/m)

REVIEW PROCESS COMPLIANCE (RPC)

Q2: Was the document read sufficiently? (0-5)

0 - None of the reviewers completed more than half
1 - None of the reviewers completed all the reading
2 - Less than half of the reviewers completed all the reading
3 - Half of the reviewers completed all the reading
4 - More than half of the reviewers completed all the reading
5 - All reviewers completed all the reading

Q2.1 What percentage of the document did each reviewer complete? (in %)

REVIEW PROCESS COMPLIANCE (RPC)

Q3: How effective were the reviewers? (0-5)

0 - Completely ineffective

1 - Major flaws in roles or categoration of comments

2 - Not effective enough for the review to pass

3 - Effective enough to be acceptable

4 - Effective

5 - Extremely effective

Q3.1: Were the correct reviewers chosen for the review? (0-3) (i.e., were all the correct perspectives represented?)

0 - none of the correct perspectives were represented

1 - At least half of the correct perspectives were represented

2 - All but one of the perspectives were represented

3 - All the perspectives were represented

(Con't)

REVIEW PROCESS COMPLIANCE (RPC)

Q3.2: Were all the meeting roles performed correctly? (0-5)

0 - Clear none followed their roles

1 - Not sure of roles followed

2 - Signs at least one reviewer followed the role

3 - Signs that most reviewers followed their roles

4 - Clear indications that most reviewers followed their roles

5 - Clear everyone followed their roles

Q3.3: Were the comments categorized validly? (0-5)

0 - There exists information but no information was recorded

1 - Some comments not recorded

2 - Comments written down but not categorized

3 - Comments written down but some not categorized

4 - Comments written down and all appear to be correctly categorized

5 - Clear that all comments were categorized correctly

(Con't)

REVIEW PROCESS COMPLIANCE (RPC)

Q3.4: What percentage of the reviewers attended the meeting? (%)

Q4: Was the meeting disposition correctly called? (y/n/m)

References

Jalote, Pankaj, Integrated Approach to Software Engineering, Springer

V. Basili, M. Daskalantonakis, and R. Yacobellis, “Technology Transfer at Motorola,” *IEEE Software*, vol. 11(2): 70-76, March 1994 .