Dependable Distributed Systems Master of Science in Engineering in Computer Science

AA 2022/2023

LECTURE 17.1 - OVERVIEW ON CAPACITY PLANNING

Part of the material inside is taken from slides of profs Menascé, Almeida, and Ciciani

Recap dependability

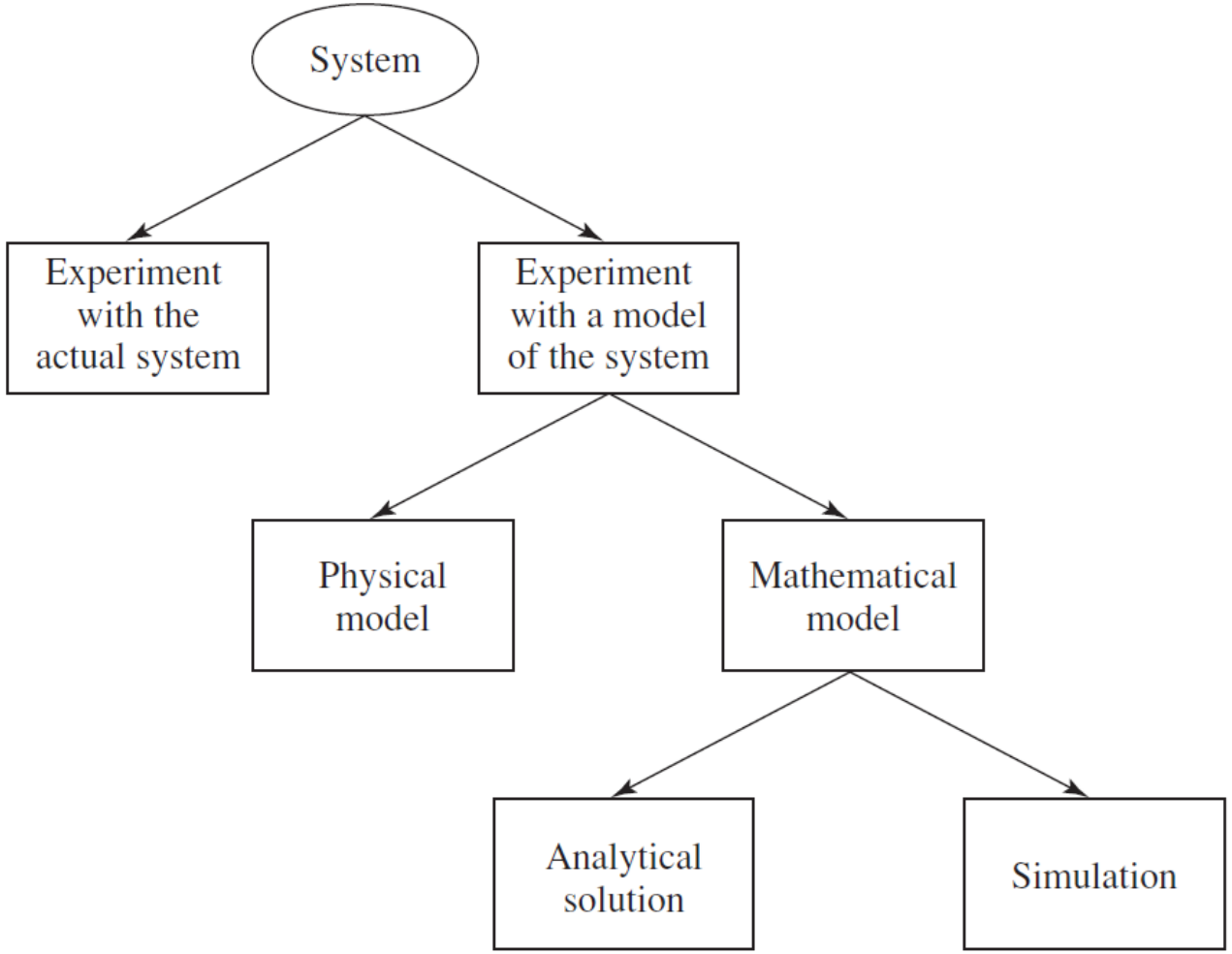
Dependability is the ability of a system to deliver a service that can justifiably be trusted,
it is the ability to avoid service failures that are more frequent and more severe than is acceptable

A **service failure** is an event that occurs when the delivered service deviates from correct service

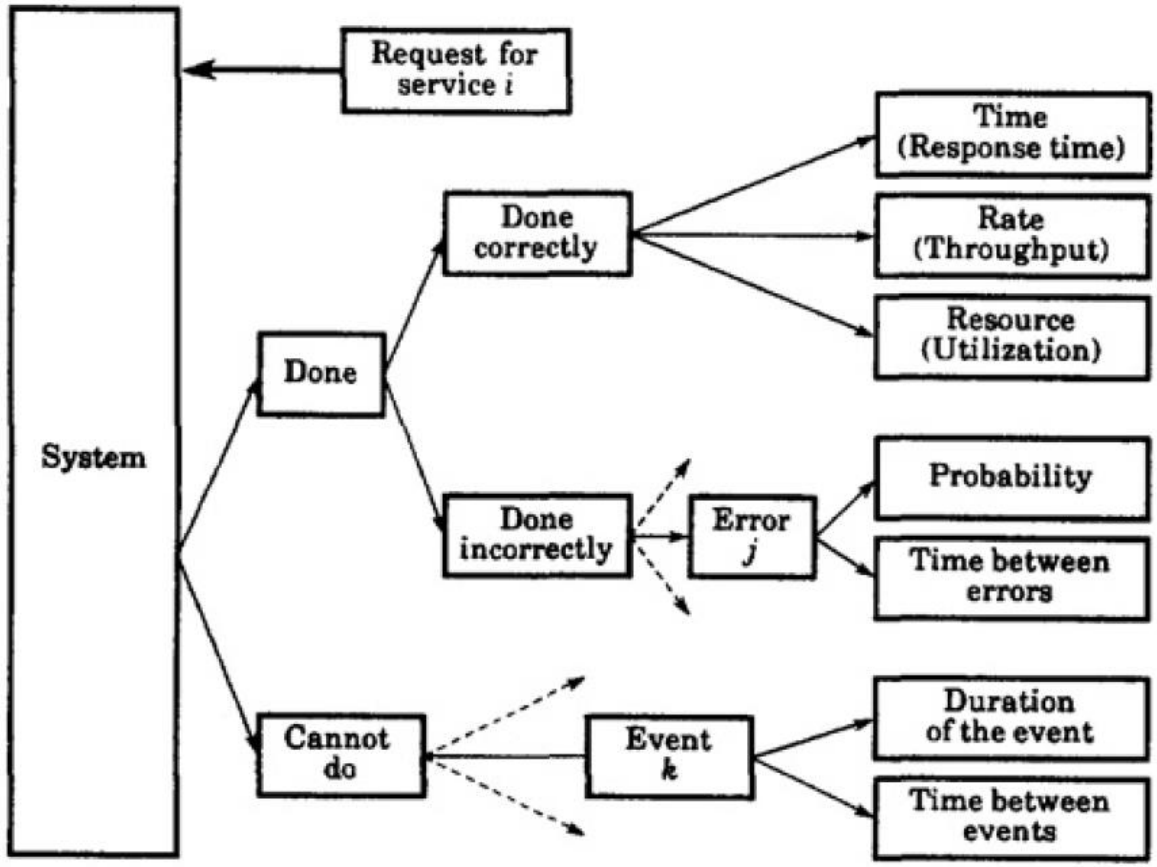
A **correct service** is delivered when the service implements its functional specifications in terms of

- **functionality**
- **performance**

Ways to study a system



Very Basics for Dependability Evaluation



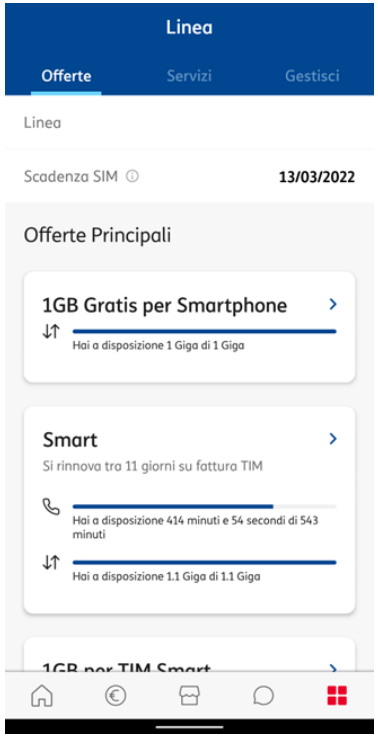
Why do performance affect correct service?
SLA

It defines how a service should operate within agreed-upon boundaries

SLAs determine what a user of an application can expect in terms of response time, throughput, system availability, reliability, etc.

- focus on metrics that users can understand
- set easy-to-measure goals
- tie IT costs to your SLAs

Why do performance affect correct service?
Users expectation



Users expectation **varies depending on what type of application** they are using and even what portion of the application they are interacting with

How do we analyze performance?

Benchmarking the system:

- limited number of testable scenarios
- potential expensive

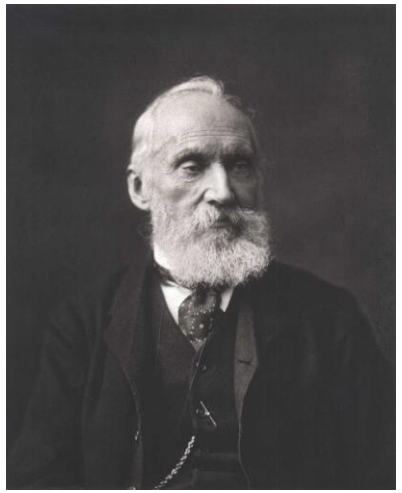
Building models

- is it possible to characterize the system and its load through models?
- cheaper

Main requirement: collect measure

The need of metrics

"In physical science the first essential step in the direction of learning any subject is to find principles of numerical reckoning and practicable methods for measuring some quality connected with it.
I often say that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely in your thoughts advanced to the state of Science, whatever the matter may be."
[PLA, vol. 1, "Electrical Units of Measurement", 1883-05-03]



Lord Kelvin

You cannot manage what you cannot measure!

How are certain levels of performance achieved? **Capacity planning**

IT capacity planning consists in **estimating** the storage, hardware, software and connection infrastructure **resources required over some future period** of time to **correctly support service provisioning**.

Alternatively

IT capacity planning is the process of **predicting when the service levels will be violated as a function of the workload evolution**, as well as the **determination of the most cost-effective** way of delaying system saturation.

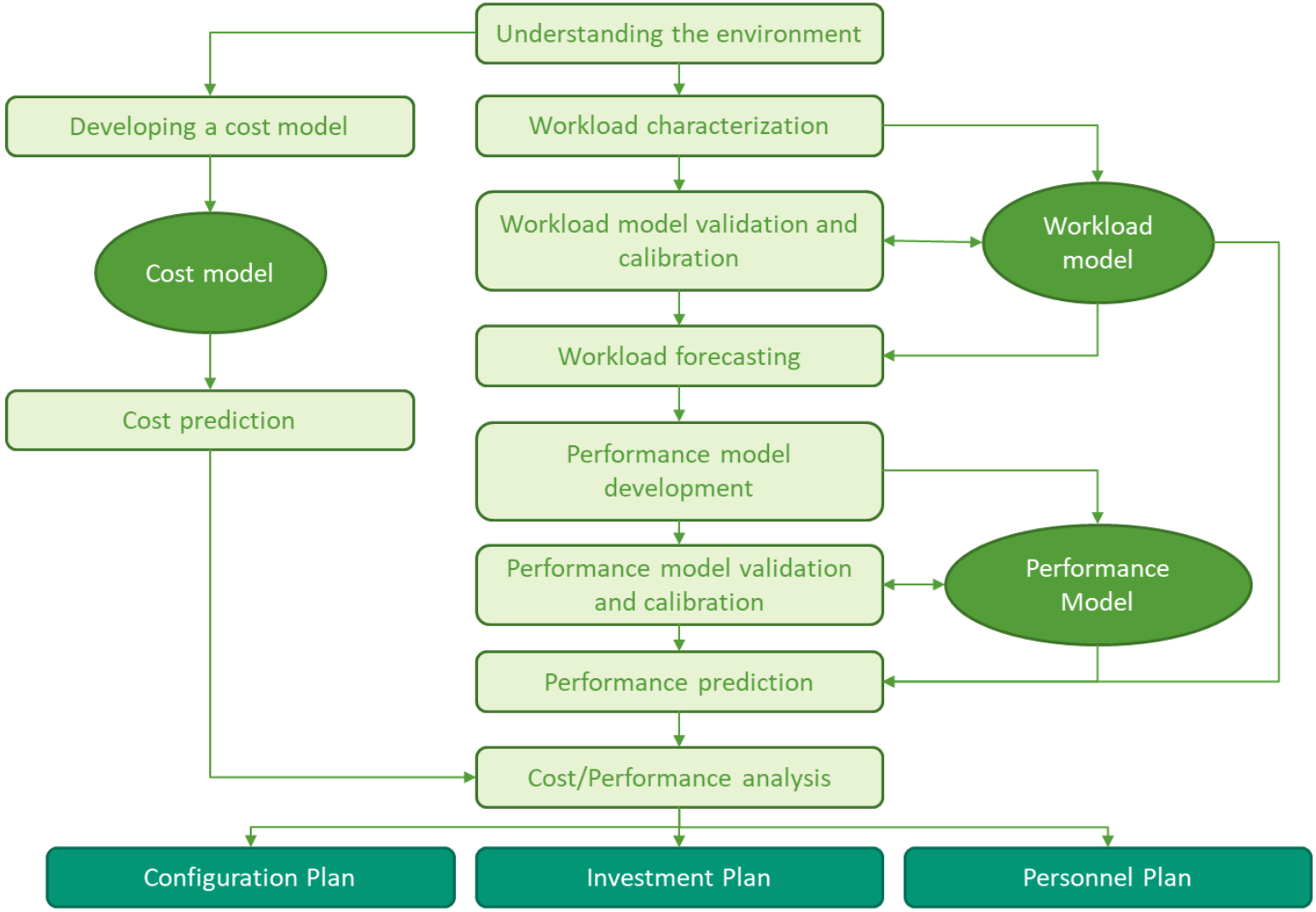
_> **Adequate capacity**

Properly handle peaks and average behavior

Why perform capacity planning?

- Avoid financial losses
- Ensure customer satisfaction
- Preserve company’s external image
- **Capacity planning problem cannot be solved instantaneously**

A methodology for capacity planning



Understanding the environment

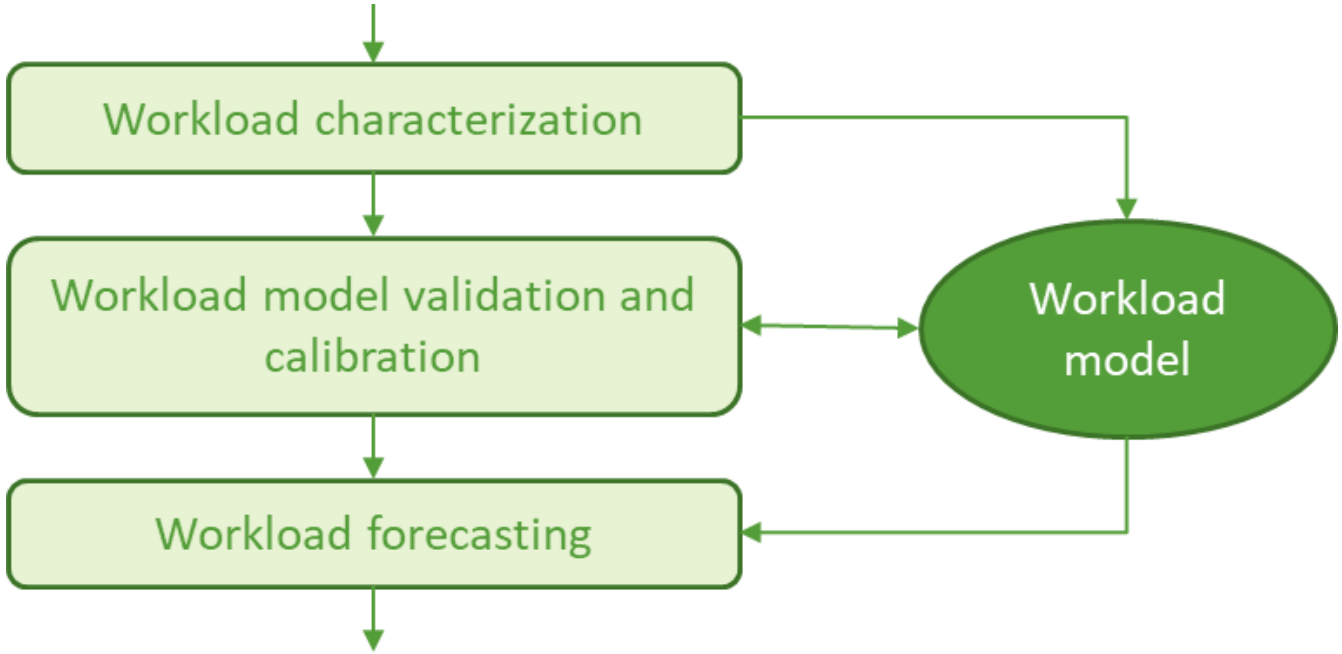
The goal is to learn what kind of

- **hardware** (**clients and servers**)
- **software** (OS, middleware, applications)
- **network** connectivity and protocols
- **SLA**
- … (whatever may have an impact on the considered performance metrics)

are present in the environment

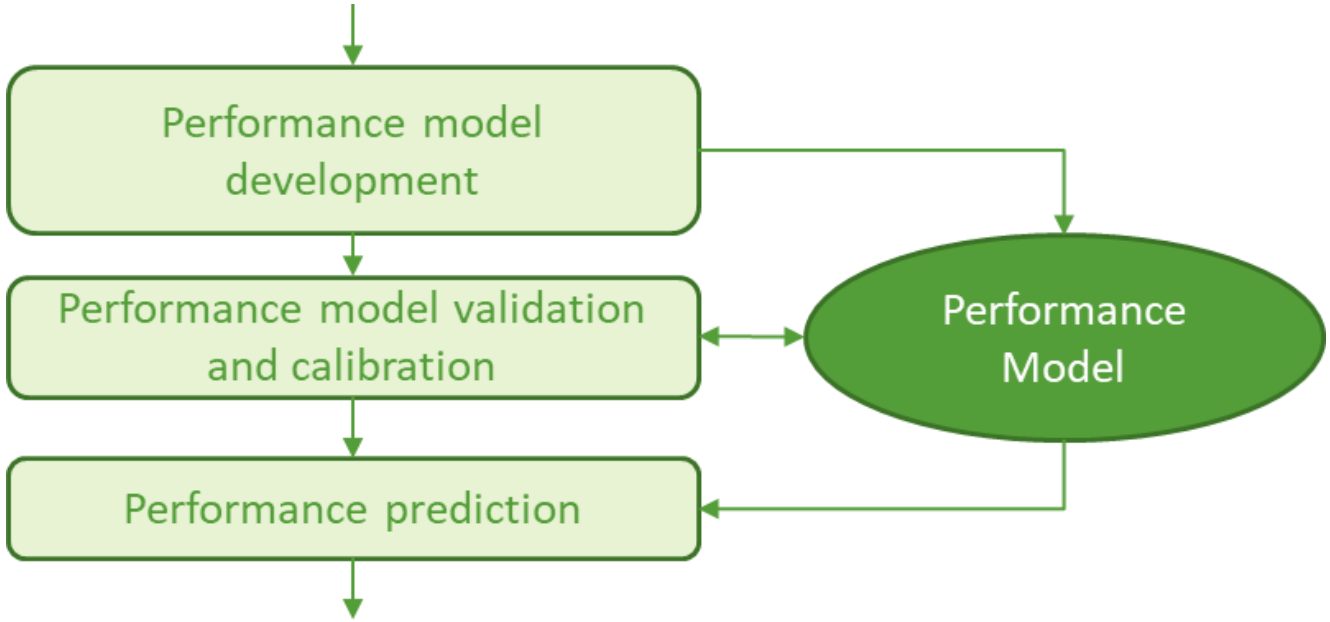
Workload model

The **workload** of a system is **the sets of all the inputs** that the system receives from its environment during any given period of time



Performance model

Used to predict performance as function of system description and workload parameters

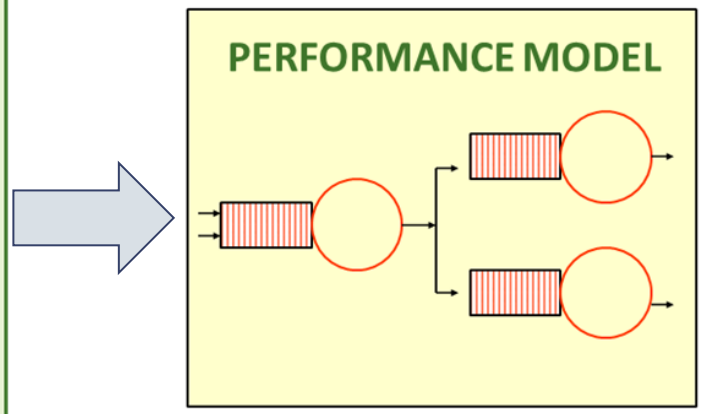


Estimates performance measures of a computer system for a given set of parameters

Outputs: response times, throughputs, system resources utilizations, queue lengths, etc.

Estimating performance measures

- **System** parameters
- **Resources** parameters
- **Workload** parameters
  - service demands
  - workload intensity
  - burstiness



- Response time
- Throughput
- Utilization
- Queue length

Parameters affecting performance metrics

System parameters examples:

- load-balancing disciplines
- network protocols
- max. num of connections supported
- …

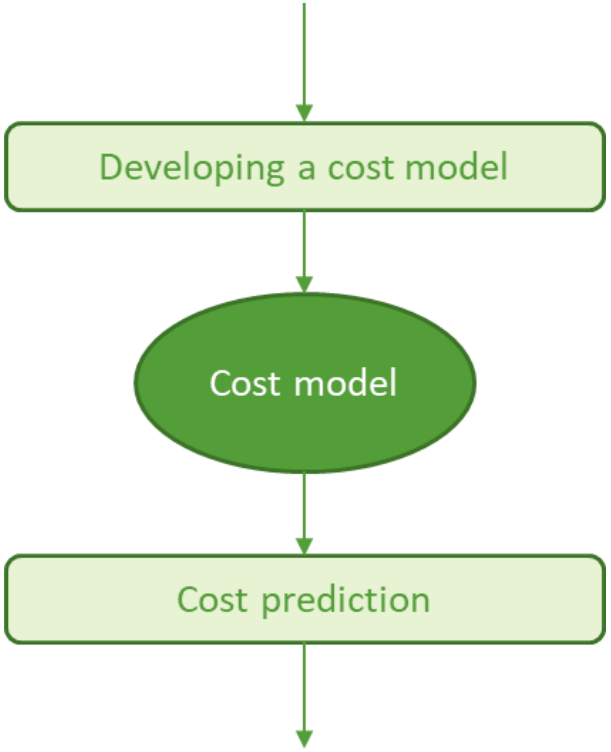
Resource parameters examples:

- disk latency, transfer rate
- network bandwidth;
- CPU speed
- …

Workload parameters examples

- WL intensity parameters:
  - num. of requests
  - num. of clients running an application
  - Burstiness
  - …
- WL service demand parameters:
  - CPU time per request
  - Disk usage per request
  - …

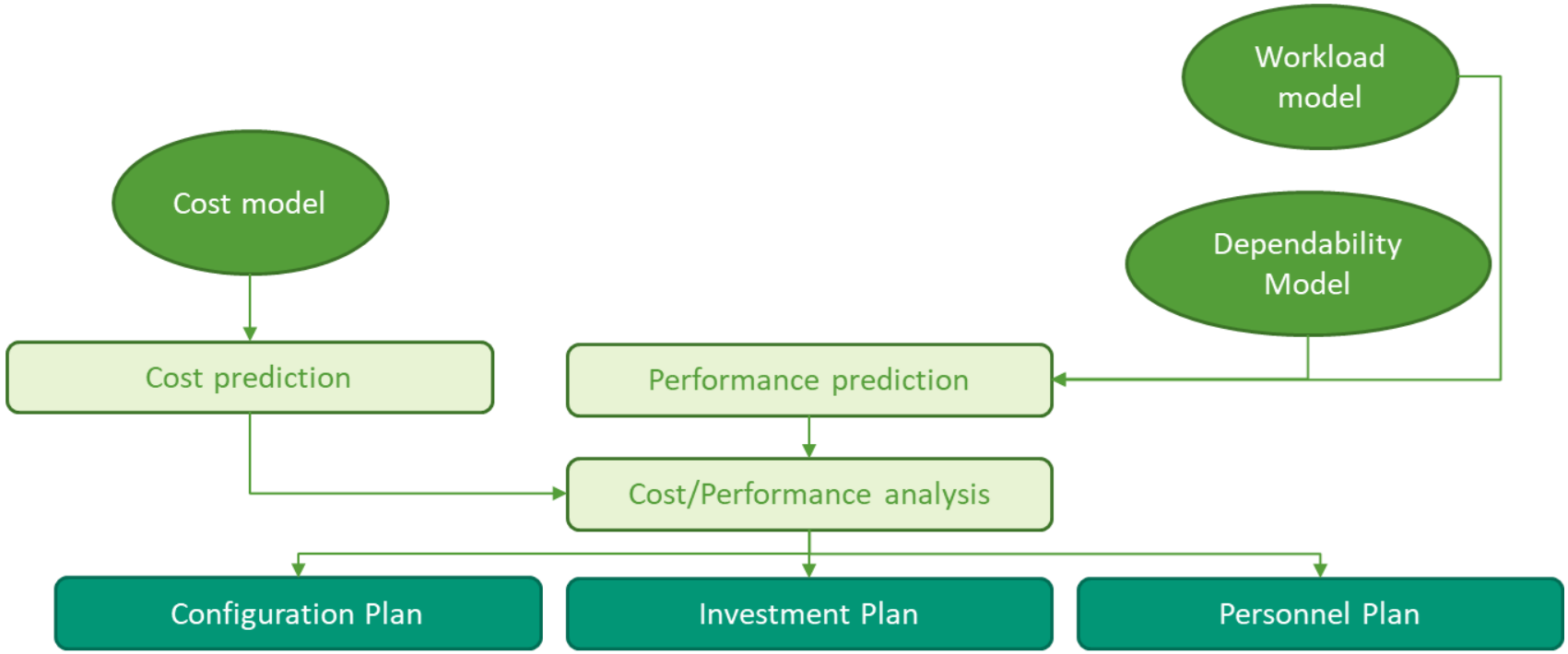
Cost model



Categories:

- Hardware cost: machines, disks, routers, etc.
- Software cost: operating systems, middleware, etc.
- Telecommunication cost
- …

Cost/performance analysis

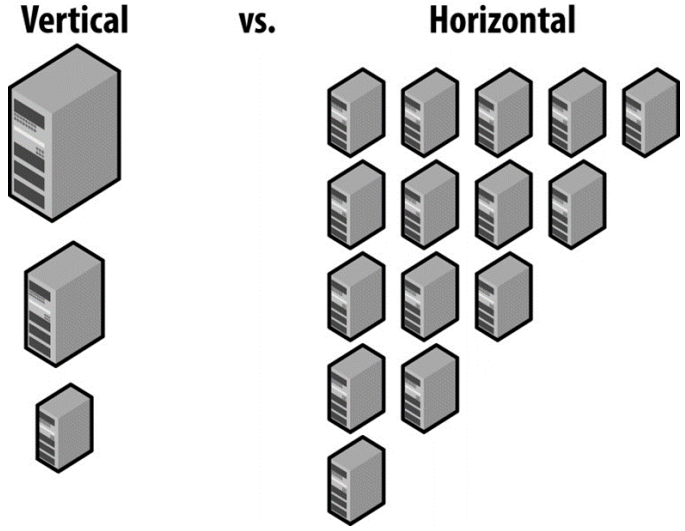


- Assess possible **scenarios**
- For each scenario, **predicts performance metrics and costs**
- Comparing scenarios, **get** configuration, investment and personnel **plans**
- Assess **payback**: ROI (return of investment), company's image, etc.

Scalability: Vertical VS Horizontal

It is the ability of a computer application or product (hardware or software) to continue to function well when it (or its context) is changed in size or volume in order to meet a user need.

- Less complex
- Upgrade limitations
- Single point of failure



- Increased complexity
- No limit to the number of processes
- Increased resilience and fault tolerance
- Horizontal scale ⇏ Increase in performance metrics

Reference

- Chapter 5 - D. A. Menascé, V. A. F. Almeida: *Capacity Planning for Web Services: metrics, models and methods*. Prentice Hall, PTR
(Available in the library inside Dipartimento di Ingegneria informatica, automatica e gestionale Antonio Ruberti)