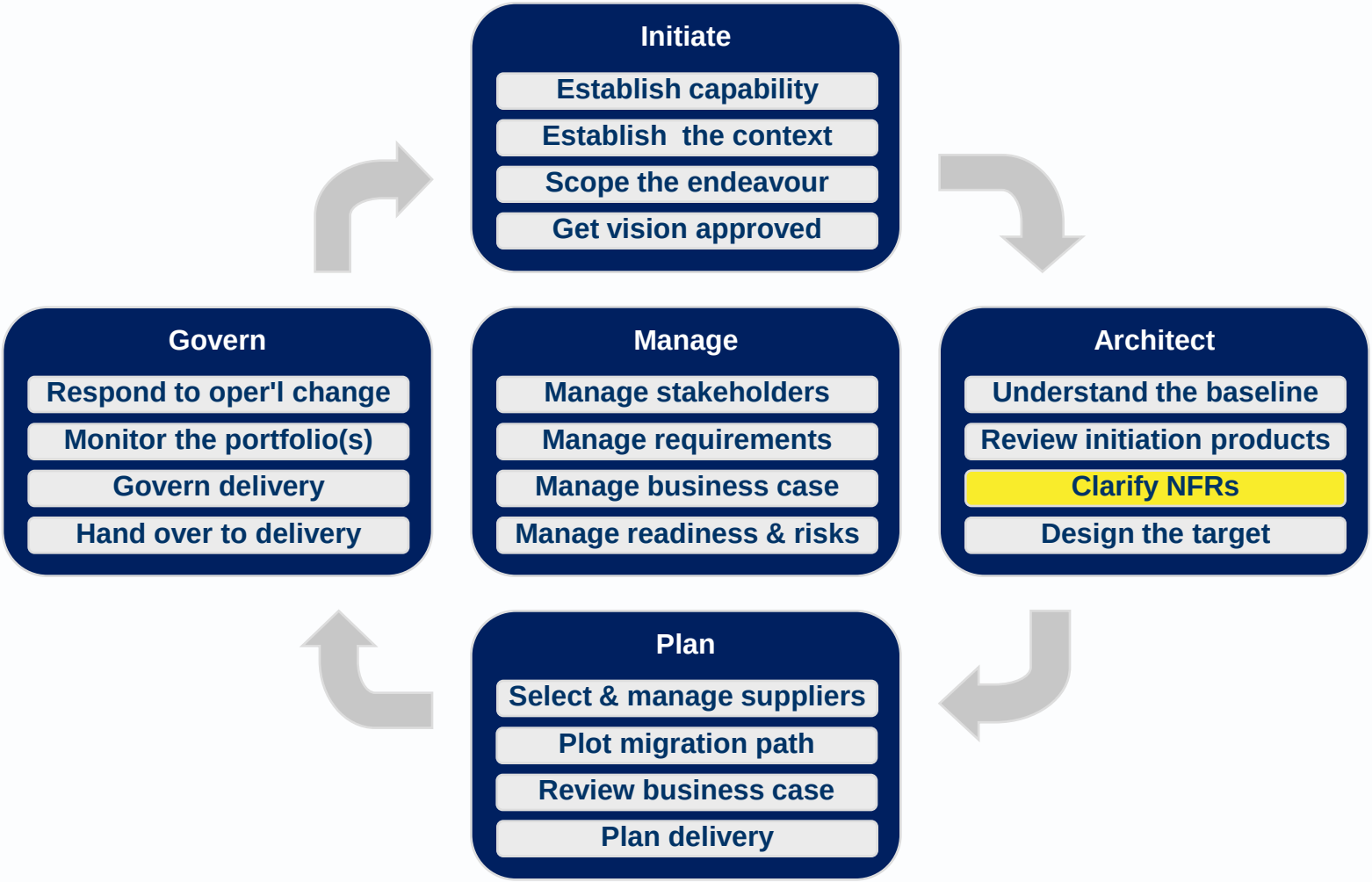
Avancier Methods (AM)
Solution Architecture

Clarify Non-Functional Requirements

CONTEXT

- What is the AM level 2 process?
- Which domain are we working in?
- What is the AM level 3 process?

AM level 2 process



What is the AM level 3 process?

Avancier

1. Set target measures with flexibility
2. Assess business & IT readiness
3. Assess non-functional risks

Use case description

Avancier

Service	Name: Book train seat
	Inputs: Journey facts
	Outputs or results: Seat reservation
Process flow	1. Identify journey start and end stations 2. Identify outbound start time 3. Identify inbound start time 4. Identify traveller numbers and ages 5. Review booking details 6. Enter payment details 7. Create personal account (optional) 8. Confirm payment details 9. Collect receipt
Non-functionals	Response time
	Throughput
	Availability
	etc

Don't forget the NFRs

Qualities include

- Performance (timings and volumes of business done)
- Availability (derived from Reliability and Recoverability)
- Integrity
- Security
- Scalability
- Serviceability
- Usability
- Maintainability
- Portability
- Interoperability
- Extensibility

Vital

- Meeting a few NFRs can vastly exceed the cost of meeting many functional requirements

- Some NFRs are critical
  - can mean the difference between success and failure of a project
  - if business operations fail, it may cost millions of dollars.

General advice on setting NFR measures

Avancier

1. Set targets that are realistic
   - Zoom out (to the business context) to assess if they are truly necessary
   - Zoom in (to the design) to assess if they are probably achievable.

2. Define how they will be measured

3. Adjust targets with a level
   - a % of cases that must meet the target, and periods when the targets will be relaxed.
   - E.g. Not ALL Telco top-up transactions must be faster than 2 seconds
     - 99% of Telco top-up transactions must be faster than 3 seconds
     - 95% of Telco top-up transactions must be faster than 2 seconds
     - If a service is expected to be available during weekdays at 99.95%, any outage at weekends doesn't impact the figure.

Pre-empt glib requirements

- Strive to steer non-functional requirements
- It is easy for customers to overstate requirements
  - and thereby impose excessive costs on solution development.
  - are well-nigh impossible to measure, or to meet,
  - are of questionable business benefit.

- Turn NFRs as early as possible into solutions you can demonstrably deliver.
  - E.g. For **maintainability**
  - you might define the use of specific “open” standards, and traceability of requirements to solution elements.

Consider also cost and risk

Avancier

- It ain't just the target measure users want
- It is what you have to do/pay to meet it

- What is the cost/risk to the business?
- What is the cost/risk to the supplier?
- And what you have to do/pay when the measure isn't met
  - Service credit
    - Money
    - Human resources - consultancy
    - Technology resources - equipment upgrade / discount

- Length of Telco down time > loss of customers
  - (known to the "Churn" department)

Performance = time and/or volume

Avancier

- Time (depends on speed)
  - Duration
  - Response time,
  - Latency
  - Cycle time

- Volume (amount of business done per hour, minute or second)
  - Throughput
  - Capacity

Trade off: high throughput or fast response time?

- "Many assume the default Oracle 11g parameters are right for them, not realizing the incompatible goals of optimizing for throughput vs. fast response time.

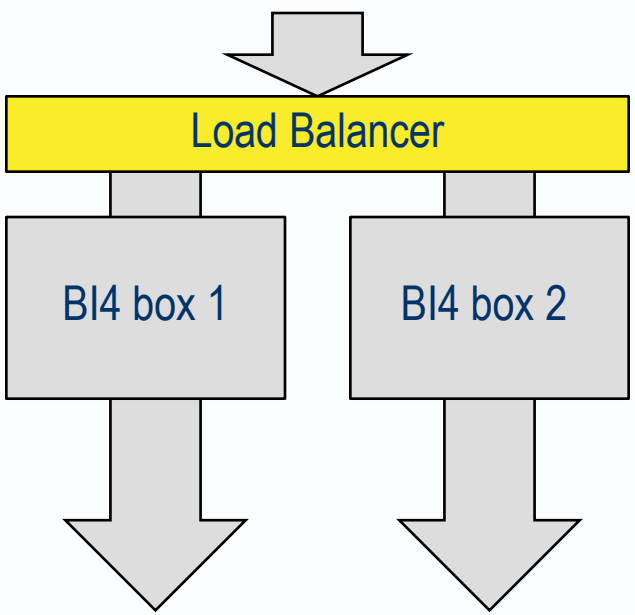
- *optimizer_mode = all_rows* (the default)
  - **maximizes throughout** (by choosing SQL access plans that minimize server resources)
- *optimizer_mode = first_rows*
  - optimizies for the **fastest response time**

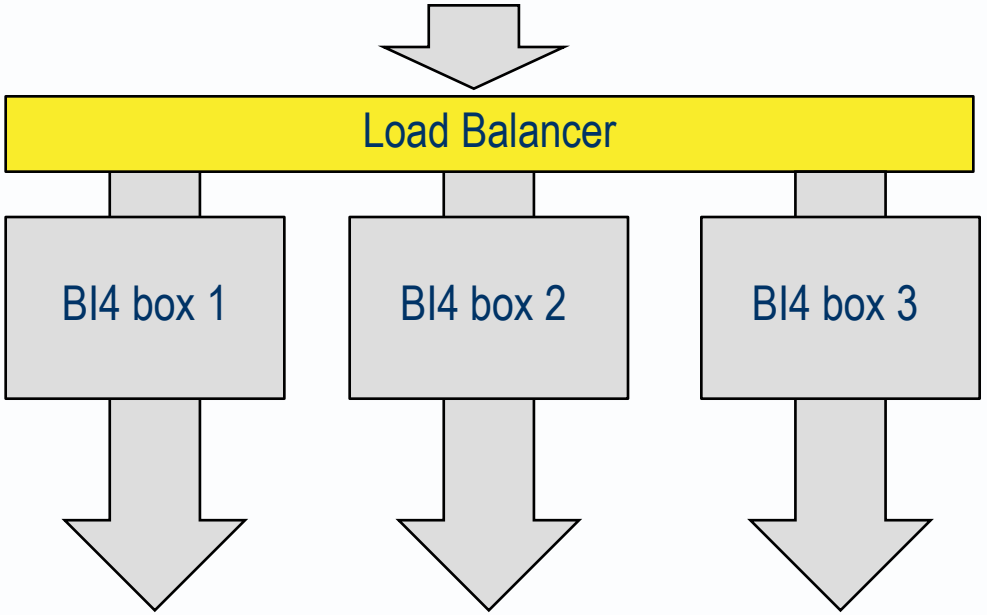
- These goals of response time and throughput are different and often at odds with each other"

- *IT Tips by Burleson Consulting, September 3, 2009*

N + 1 Design

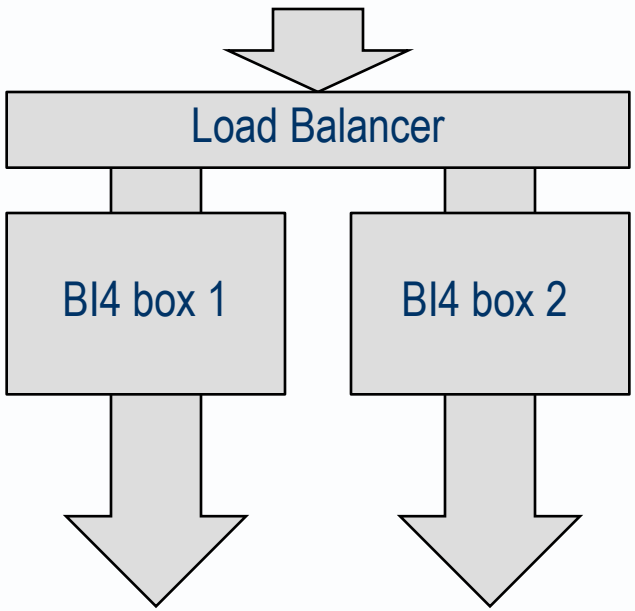
- Add 1 to the number you think you need – in case one fails

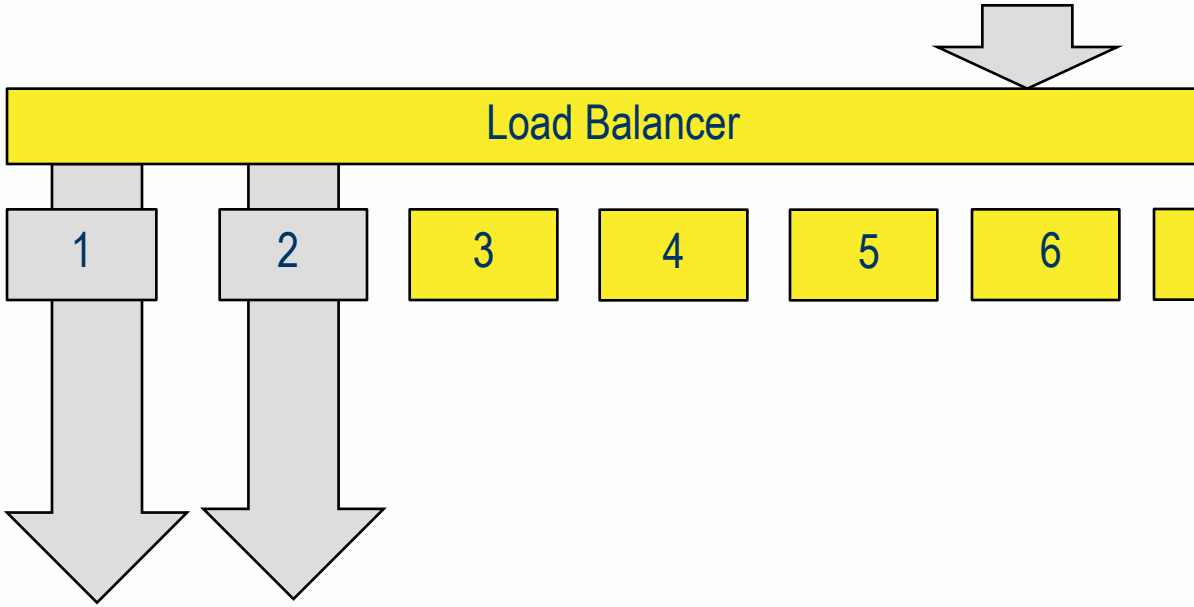




Scaling out from N+1 design to N+200 design?

- "WebI on HANA scales in a linear manner w. number of users"
- Are you sure?





Themes of Avancier's architect training

- 1) Think about the business context
- 2) Don't forget the numbers
- 3) You have to balance trade offs.

Trade off: flexibility versus simplicity/speed

- Designing for flexibility (to facilitate meeting future requirements) usually implies

1. Anticipating now what *kind* of requirements will emerge
   - You cannot design now for changes you cannot envisage

2. Paying the price now of
   - Extra complexity
   - Extra cost
   - Slower performance of the first-released system

Open the “Design for NFRs” chapter 14

Avancier

- Look for the short tables under each section heading

Key drivers for architects in solution definition

Remember PARRISSS	Requirements that you may well have to measure in a reasonably standard way
Performance	Measure: response or cycle time. Measure: throughput (e.g. transactions per minute)

Business continuity measures

Avancier

Availability

Measure: the time a service is required to be usable during the period (window) that it should be usable.

A fraction, normally expressed as a percentage.

May be calculated *for a single component* as:

Reliability / (Reliability + Recoverability), or

MTBF / (MTBF + MTTR).

Somehow derived from

Reliability

Measure 1: the duration for which a service or a component is required to perform under specific conditions without failure.

This may be presented as **MTBF** = Mean Time Between Failures.

This measure normally applies to the total failure a component, rather than a transient failure to respond (which may be accounted for in response time measures).

OR

Measure 2: the percentage of successful service invocations. E.g. if 100 attempts at using a service result in 2 time-outs and 1 data corruption error message, the service can be said to be 97% reliable.

Recoverability

Measure: composed of the first or both parts below:

RTO (Recovery Time Objective) or **MTTR** (Mean Time To Repair).

The Duration to restore a service back to its operational state after a failure

RPO (Recovery Point Objective) the maximum age of data input up to the failure that can be lost.

Data integrity and security measures

Avancier

Integrity	Measure 1: data store: a percentage of records with integrity errors (missing data, violated primary/foreign key relationships, etc.). Measure 2: data flow: detected incidents of data flow corruption. Measure 3: reported level of trust in the integrity of the system and its data.
Security	Given you know who (users, machines) can access what (data or process). Measure 1: a percentage of successful attempts to overcome the security barriers: as a Failed to Successful ratio (i.e. a secure system has a high percentage value). Measure 2: incidences of unauthorized access to components (data or programs). Security requirements may not be obvious to users. Conduct an analysis of vulnerability to check you know the security requirements. The principal threats to be countered are: •Loss of confidentiality of data •Unavailability of data or services •Loss of integrity of data •Unauthorized use of resources. •Be sure to know audit requirements.

Other measures

Avancier

Scalability	Given you know the numbers to which the system(s) may have to grow or shrink its performance or capacity, and in what time frame. Consider measures for •static (design time) scalability: Time to redesign to a specific scale. •dynamic (run time) scalability. Incidents caused by failure to scale.
Serviceability	The ability to identify and solve problems, which requires: •Manageability: the ability to gather information about the state of something and to control it. •The ability take corrective action, e.g. to repair or upgrade a component in a running system. (ITIL interprets Serviceability differently).

Don’t eat more UM PIE than you need

Avancier

Requirements that may be so difficult to measure you’d better invent your own achievable measure or rule them out.

Harder to define, prioritise & demonstrate compliance

Usability

Avancier

Don't eat more UM PIE than you need	Requirements that may be so difficult to measure you'd better invent your own achievable measure or rule them out.
Usability	Measure: well-nigh impossible to measure this objectively in a way that corresponds to uses perceptions of the system. But consider PLUME. ■Productivity ■Learnability ■User Satisfaction ■Memorability ■Error rates What is important to users? Consider •Accessibility legislation. •International operation, including multi-lingual and multi-cultural abilities. •Human Factors design (such as the flow of on-line forms) •The 'technical competence' required of the user to make use of any particular system functions. E.g. it probably should not be necessary for a 'standard' user to have to define MPEG parameters (such as bit rate, oversampling, etc.) to play an audio file. However, a 'studio engineer' user might need control over those kinds of things.

Maintainability

Avancier

Don't eat more UM PIE than you need	Requirements that may be so difficult to measure you'd better invent your own achievable measure or rule them out.
Maintainability	Measure: software change/defect frequencies and cycle times. What is important to you? Consider •How you distinguish defects from minor changes from major changes •Ease (effort needed) of defect removal, functional change, or non-functional change. •Stability - Risk of unexpected effect of modifications. •Testability - Ease (in terms of effort needed) of validating the modified software.

Three related measures

Avancier

Don't eat more UM PIE than you need	Requirements that may be so difficult to measure you'd better invent your own achievable measure or rule them out.
Portability	Measure the ease of which a particular target service (i.e. object code) can be moved to a new operating system or hardware platform. What is important to you? Consider •Adaptability - Is recoding needed? •Installability - Is installation easy? •Replaceability – Can it replace other existing services?
Interoperability	This is a design-time commitment What is important to you? Consider: •interoperability with systems outside the enterprise. (E.g. interoperable calendaring or scheduling functions may be key to the usefulness of a system.) •local interoperability standards: (networking protocols, adapter standards, service interface standards). •industry interoperability standards (OSS/J for Telco OSS components, JAIN for network control, and W3C or OASIS standards). Supplementing this, or even as an alternative, a particular architectural mandate could be defined to use EAI technologies to provide interoperability.
Extensibility	The ability of a system to increase it's coverage in areas of scope that it did not do at its inception. These areas are usually defined at design time - but systems features for those areas are not designed or implemented. The ability to increase the system's scope is usually a function of modularity and decoupling, which is what the designer would need to incorporate, so the requirements for the system would have to say something about the need to be able to add new functions after system delivery.