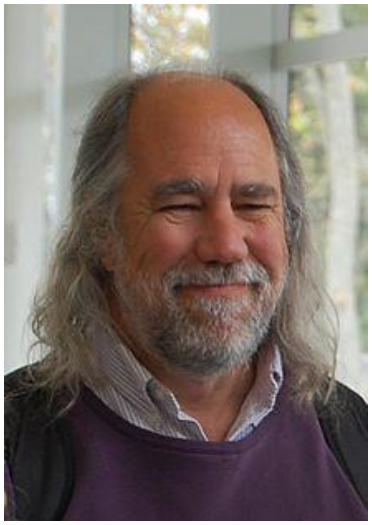
Agenda

- **Let's remember**
- **Objectives**
- **Introduction**
  - Market Trends
  - Real Cases
- **Functional vs Non-Functional**
- **Re-engeneering the Requirements**
- **Non-Functional Requirements**
- **Let's play ☺ - Real case analysis**
- **Experiences from projects**
- **Conclusions**

Let's remember



Grady Booch

It is a privilege to be software developer, because we change the world.

It is a responsibility to be software developer, because we change the world.

ICSE 2015 - Grady Booch Keynote (1h23m)

2

Why NFRs

Functional vs Non-Functional

Functional requirements define *what* a software product must do: its features and functions

Non-Functional requirements specify *how* the system must perform

What requirement is it?

"A user must be able to edit messages after they are sent."

"The message must be updated for all users in a chat within 0.1 seconds, given that all users are online and have LTE connection or better"

Objectives

Understand the Non-Functional Requirements:

How to define them

Metrics to consider

Impact on End Users

Quality Assessment

What if... We miss to define NFRs

Do you think NFRs are considered / impacting Business Solutions today?

Some Inputs from Market

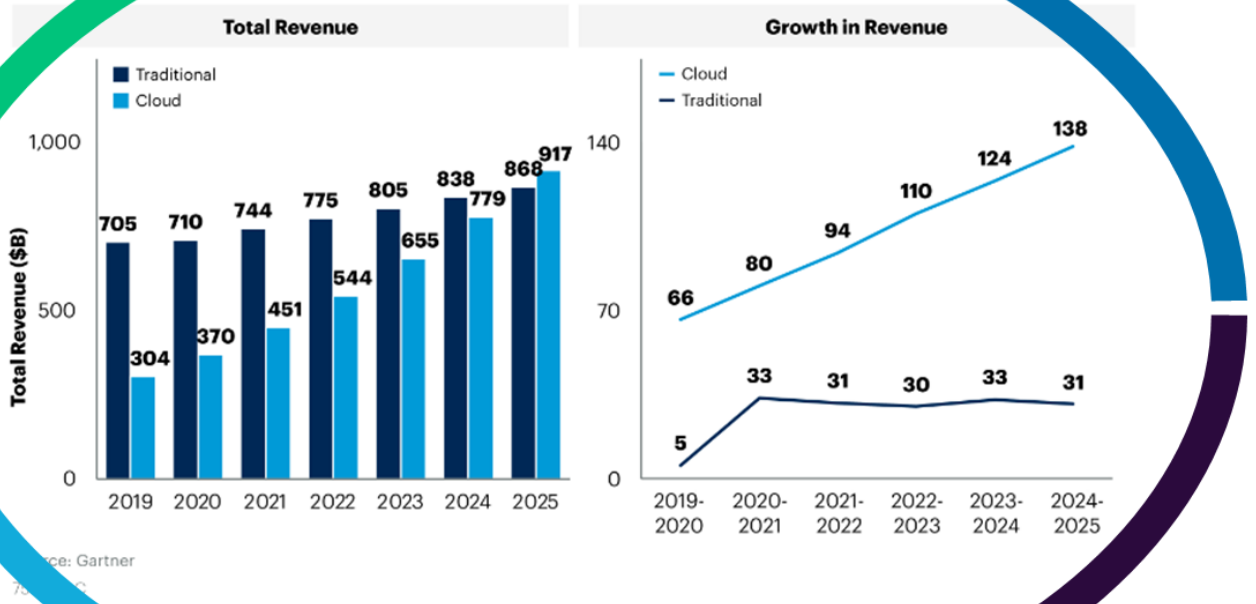
NFRs are key in Modernization! Modern system architecture and what it means to quality

Trends

Post-Pandemic reality

Market trends

Competitivity



Sizing Cloud Shift, Worldwide, 2019 – 2025
According to Gartner

Shared Resources

Leaving the On-premise

Real Cases

The Insurance Insider
86.268 seguidores
1 año

Lloyd's shuts down online systems in precautionary action
The Corporation is resetting systems as it investigates "unusual activity" on its network.
http://spr.ly/6043Mdq5W #InsuranceInsidernews #insurance #reinsurance

@Fatima__NB · **Seguir**

@Renfe empieza su nuevo proyecto @Avlorenfe con la pág de registro caída 40min antes y al estrenar avlorenfe.com se ha colgado por la peticiones masivas. De verdad tenéis tan poca previsión? Esto no es un buen comienzo y deja mucho que desear del futuro "servicio" #AVLO

1:11 p. m. · 27 ene. 2020

6 Responder Compartir

Leer 1 respuesta

HT TECH
MAKE LIFE SIMPLE

Search for news, mobiles, laptops etc..

HOME NEWS MOBILE LAPTOPS PC GAMING GADGETS RECOMMENDER COMPARE PHOTOS VIDEOS

All three major services of Meta Platforms, Facebook, WhatsApp, and Instagram are facing outage, keeping as many as 13,000 users from accessing the apps. (REUTERS)

WhatsApp, Facebook, Instagram down: All three major platforms of Meta, Facebook, WhatsApp, and Instagram are experiencing major outages in the US, as per reports. More than 13,000 users of these apps have reported various issues with accessing the apps and the web interface. The issue has affected the users in the US region and has not affected the rest of the regions, based on the information.

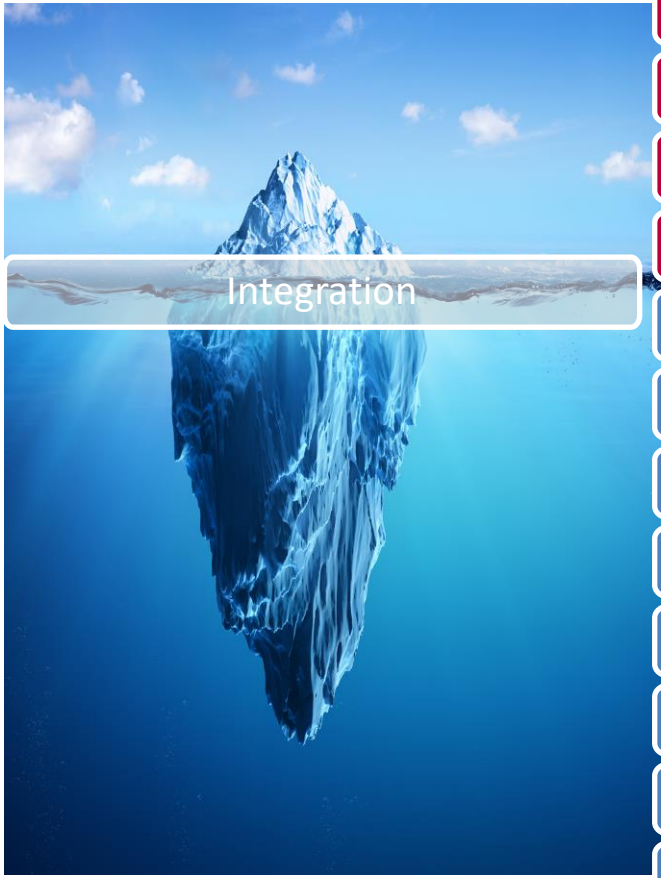
FALLO INFORMÁTICO

Cataluña suspende el requisito de certificado covid por un colapso del sistema

En las últimas horas el sistema de La Meva Salut se ha colapsado y muchos usuarios no pueden descargarse el código QR

Un camarero comprueba el pasaporte covid de un cliente. (EFE/ Marta Pérez)

Functional vs Non-Functional



- End To End Use Cases
- End Users' Behavior
- System's Properties according to User's Requirements
- End User's Profile
- Business Flows
- Non-Functional Requirements Assessment
- Recovery
- Behavior under load
- Security
- Additional external components
- Data Model
- Infrastructure

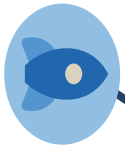
Non-Functional Requirements

Performance	How fast a system should respond to requests
Scalability	How well a system can handle an increase in users or workload
Security	How well a system protects against unauthorized access and data breaches
Usability	How easy a system is to use
Maintainability	How easy it is to update and modify the system

Accessibility

Sustainability

Re-engineering the Requirements



Setup & Checkout

Code is checked out by developers and built in the selected environment as per setup file

Unit Tests

To ensure correct interoperability between the modules

Static Analysis of Code

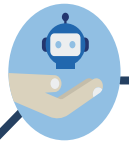
The relevant static analysis scans and security scans are run against the code base to ensure that certain coding and security best practices have been adhered to.

Deployment to Test Environment

Code is deployed into an integrated test environment where it will exist against other modules (actual deployed code sitting outside of the current repos domain).

Packaging & Deployment

Only after the involved / selected Quality Gates are identified as passed



Automated Functional Tests

Automated tests identified by the testing team are executed. This will span a wider coverage of the codebase and will include real users' behavior tests, with more realistic data that better resembles production.

Dynamic Code Analysis

Run against live code (unlike the static analysis scans which are run against pre-deployed code) to provide an additional measure of quality and security checks.

Load & Performance Tests

To confirm that the code does not cause any type of potential or real bottleneck or performance degradation from end users' perspective

Non-Functional Requirements

Security & Compliancy

- API level and Application level security vulnerability requirements
- GDPR compliancy
- Complying with industry security standards (PII, PCI DSS)

Performance

- Pro-active event-based scaling to fulfill future anticipated load - Performance load & Stress
- Validate just in Time infrastructure
- Key transaction response time within SLA

Availability

- Disaster recovery, Fault tolerance - Availability and speed to recovery

Quality @ Speed

- Faster development processes drive a need for testing at speed to support release cycles (Devops & CT)
- Manual testing delays delivery

Integration & Interoperability

Current systems involve components Integration

- Application service exchange data with other SaaS & PaaS Solution
- Application service response time within SLA
- Denied Unauthorized service access

Accessibility & Portability

- Applications should be validated on across Browsers, Mobile devices & OS
- SaaS solution access by different channels & Devices.
- To setup as well as validate functionality on Multiple environments is challenging

App Modernization

- Validate Application Modernization – Extending legacy application to Utilize multiple service providers (Cloud?) and scale

Data Handling

- Validate data transition and model
- Application Configuration to handle different access levels to Data
- Application can be accessed via all agreed channel

NFRs - Inputs

- What is the system architecture and how is it built?
- How many end users will actually use the application?
- Where are these end users located?
- How many of these end users will use it concurrently?
- How will the end users connect to the application?
- How many additional end users will require access to the application over time?
- What will be the environment under test and what relationship (in terms of capacity) will it have with Production?
- What effect will the application have on key servers' consume of resources?

Let's Play

Let's consider Ticketing Application, what are our Requirements?

Type of Requirement	Impact	StakeHolder	Metric
Functional			
Non-Functional			

Real examples NFR Assessment

Context

Business Key application, in order to find a more efficient resources consume, **was re-designed to fit cloud infrastructure**

Challenge

Code was developed and progressively validated, which **forced to coordinate efforts**

Objectives

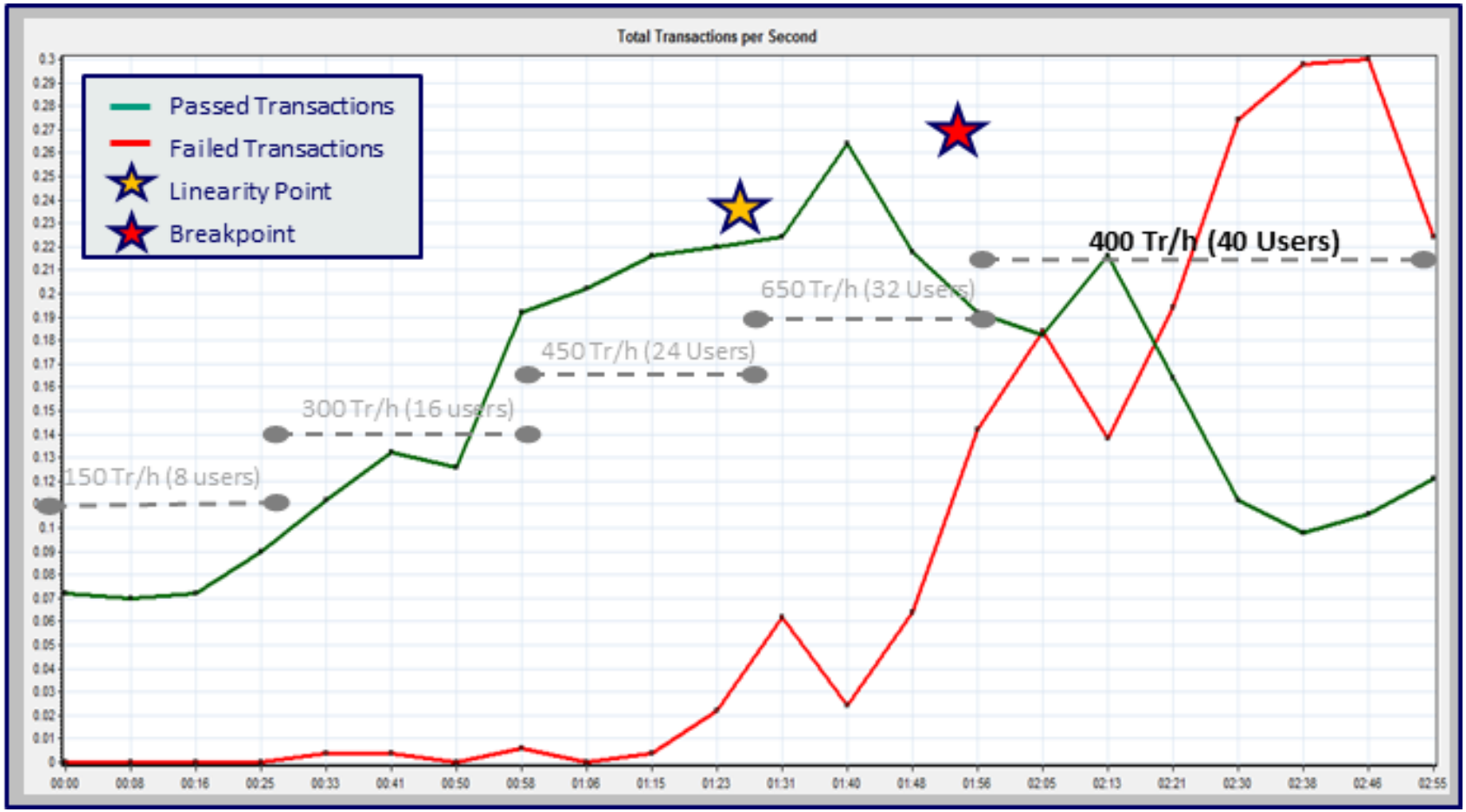
Assess system's stability under high volume of users uploading and to processing data **and configure the Infrastructure**

Solution

- Detection of most **sensitive to increasing load services**
- Tuning of different **escalation parameters**

Continuous NFRs Assessment in parallel to developing and infrastructure's configuration allowed to take an initial Baseline, analyze metrics, size the infrastructure and configure scalation rules

Real examples
NFR Assessment



Non-Functional Requirements

Data Journey

Observing Data

Data-quality-related issues in:

- Completeness
  Accuracy
- Uniqueness
- Timeliness
- Anomalies and outliers

Observing Data Pipeline

Changes in data pipeline when data is transitioning between different stages:

- Schema
- Volume
- Behavior/performance
- Frequency
- Time

Observing Data Infrastructure

Logs and metrics around resource consumption:

- Compute
- Performance
- Network
- Storage
- Hardware resources

Observing Data Usage/Users

Metadata and lineage in:

- Who owns the data
- Who changed the data
- Who uses the data
- How often the user uses the data

Observing Cost & Financial Impacts

Financial impacts in:

- System downtime chargeback
- Resource consumption
- SLA engagement
- Capacity planning

Observing Data In Use

Last Mile Data Validation and Testing

- **Correct Dashboard Data**
- **Accurate Data exports**
- **Predictive Model Accuracy**

Conclusions

Application Context	Let's clarify main application's goals and the architecture we may dispose of to make it working
Identification of main business goals	Let's make sure that all the stakeholders understand not only the functionality of the application, but also there must be a full alignment of the key cases
Environments and resources	Let's ensure that we dispose of different environments and that they are properly maintained and dimensioned
Non-functional requirements	Let's make sure that, even before we start designing an application, volume of work this should handle and expected response times are clear