AGILE GS SW: CHEAPER, FASTER AND BETTER
INTRODUCTION

- **Before agile**
  - Regressions and SW failures
  - Low productivity rates
  - Lack of involvement and responsibility of development teams
  - Crisis solved by “heroes”

- **Transformation vision**
  - Be a happy team creating the best ground control SW in the world

- **Transformation objectives**
  - Improve SW quality
  - Work with peopleware values
  - Customer satisfaction: Frequent deliveries & velocity
Project-oriented groups can lead to:
- Different solutions for the same issue
- Sub-optimum workload distribution
- Priorities not properly addressed

Agile teams benefits:
- Backlog oriented to provide the best value
- Team velocity increase
- Continuous improvements in the team
Hierarchical organization consequences:
- Project manager is responsible, but
- Development is done by the team
- Team involvement is limited

Peopleware:
- SW is done by people, not by machines
- Work with motivated teams
- Focus, prioritize, complete as many tasks as possible
NEW ROLES
The challenge: to extend the Agile culture to customers

Technical staff usually open to quickly adapt the model (early adopters)

- Frequent delivery as key factor for showing the advantages of the Agile model
- Quick reception of the systems, less uncertainty
- Collaboration, development based in real needs

Higher resilience in management layers (laggards)

- Different models need to be applied in organizational and financial matters
Most customers prefer traditional **Requirements**

- Contractual: *apparently* easier to manage
- Locked/Limited

But... a **User Story** is just a well-expressed requirement

- **User language:** Communication, feedback, collaboration
- **Flexible:** Adaptive to user needs, innovation
- **Simplified planning:** Just-in-time, meet objectives
**User requirements /acceptance tests**
converted into:
- **Epics** → **US** → **Tests**

- A Docker container is provided after every Sprint
- Improved customer feedback and satisfaction

- Classical validation phases (FAT/SAT) are kept but lighter and with improved results
Project schedule based on Gantt:

- Generic – and big – tasks (Design, Integration)
- Product is only available at the end
  - Too late to complain!
- Development delays shorten testing
  - Robustness at stake

- **Frequent delivery of working SW**
  - After each sprint, SW is ready to be delivered
  - Customer can see the functionality early in the process
  - Testing is a must
    - iTeam
    - Long-term testing in parallel to quick deliveries
    - Fixes introduced as needed in the backlog
Agile is a framework for complex developments

Agile process can be stronger, integrated and coherent thanks to QA and CMMI techniques

QA system and **CMMI** as reference models with a collection of best practices
- Can be tailored to adopt the **flexibility** and **adaptability** of Agile
- Traditional waterfall model replaced by incremental approach
- Some practices are revisited to remove not useful activities
- Legacy code usually prevents **frequent delivery**
- Frequent delivery implies **automation**
- Automation applied to the **whole development process**: code audit and metrics, build, deploy, testing and validation

- Very low level, even inexistent **automatic testing** in legacy code
- **Reengineering** for introducing automation: development processes, tools and code
### Agile Success Story: EUTELSAT at GMV

#### DEVELOPMENT
- Automation
- Improved Testing
- Processes & Tools
- Fast Delivery

#### TEAM
- Training
- Self-organized
- Roles & Leadership
- Ceremonies

#### CUSTOMER
- Collaboration
- Feedback
- Flexible
- Satisfaction

#### CORPORATION
- Organizational
- Cultural

---

Applied to all development layers
- Requires reengineering and refactoring
- To support the practices at all levels
- As the key factor in the process

To understand the practices
- Based on internal and external agreements
- Versus project monitoring and control
- As the key factor in the process

To focus in the most valuable solution
- to understand actual needs
- To adapt to the project evolution
- Because of the goals obtained

All Company levels need to be involved
- This is just the beginning ...
LESSONS LEARNED

MISTAKES
- SM not defined (1 team)
- More than 1 rol per person
- Non-agile practices in agile

EXPERIMENTS
- Inception
- Grooming
- Retro dynamics
- Salary increase by the team

PRACTICES
- Automation: metrics, test
- Freq. delivery
- Stable teams
- Daily
- Timebox (sometimes)

FAILURES
- Lean coffee
THANK YOU