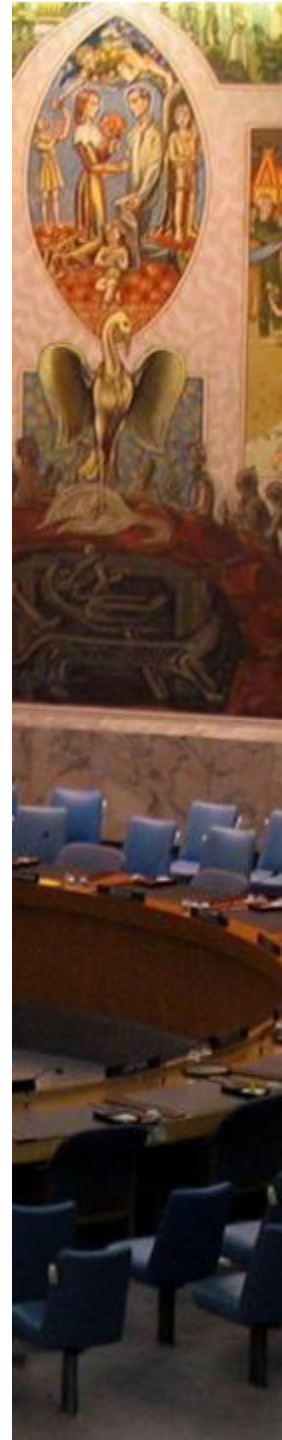
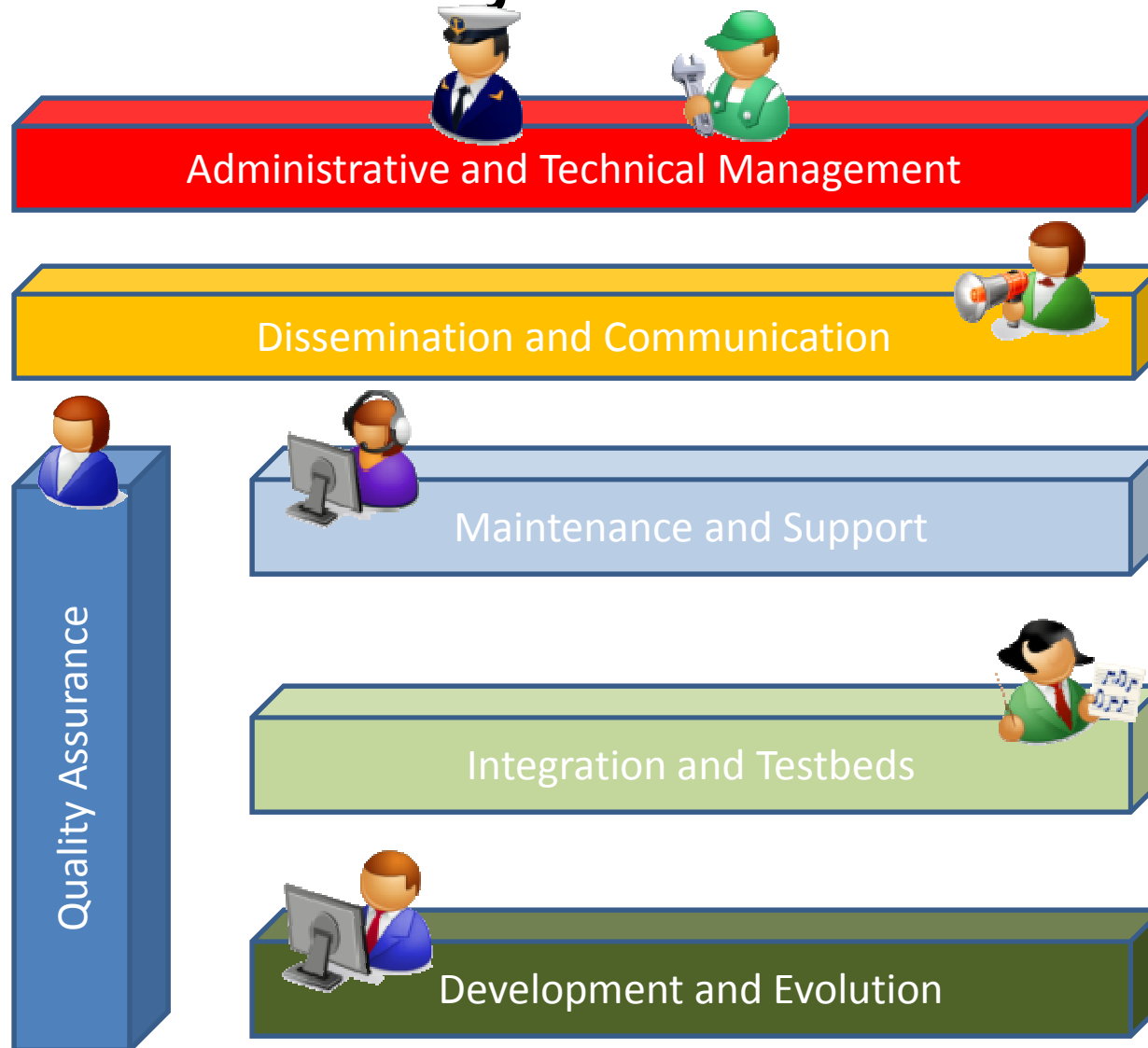




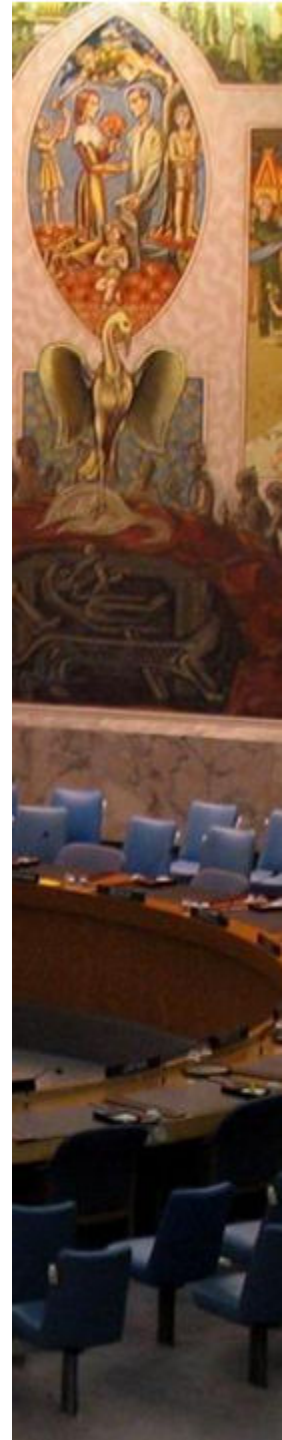
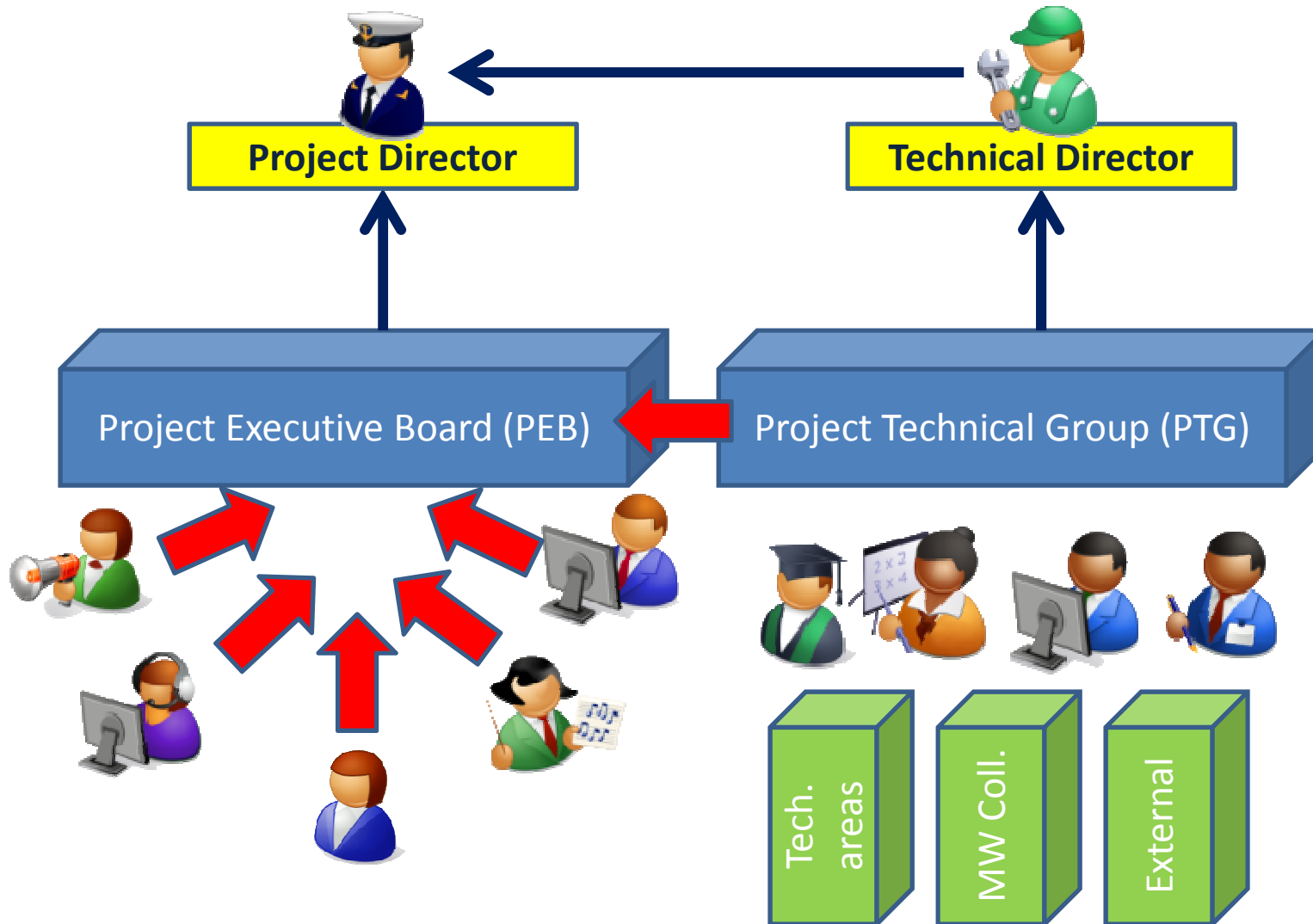
# European Middleware Initiative (EMI) The Software Engineering Model Alberto Di Meglio (CERN) Interim Project Director

# Project Structure

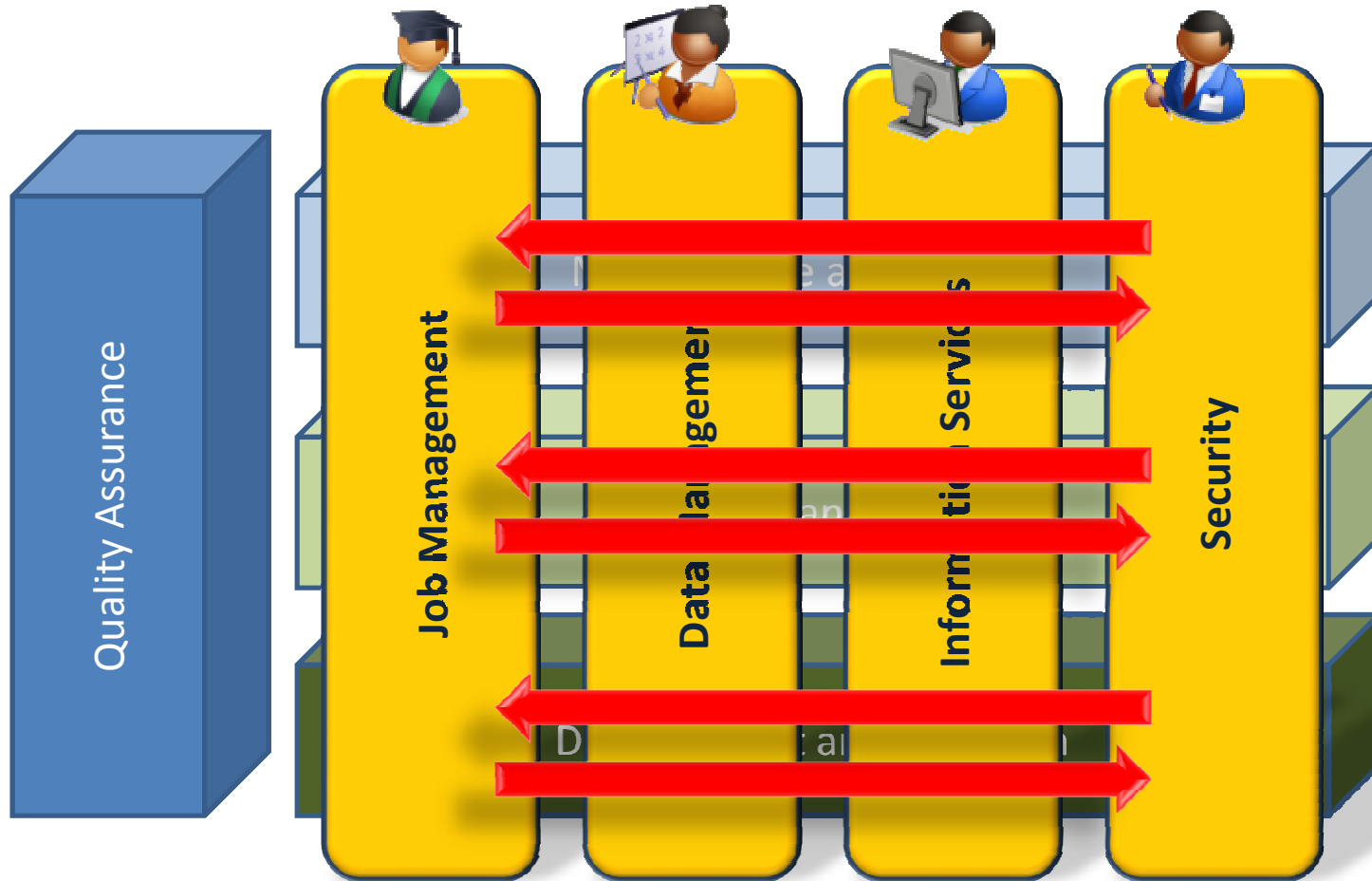




# Project Execution



# Technical Areas

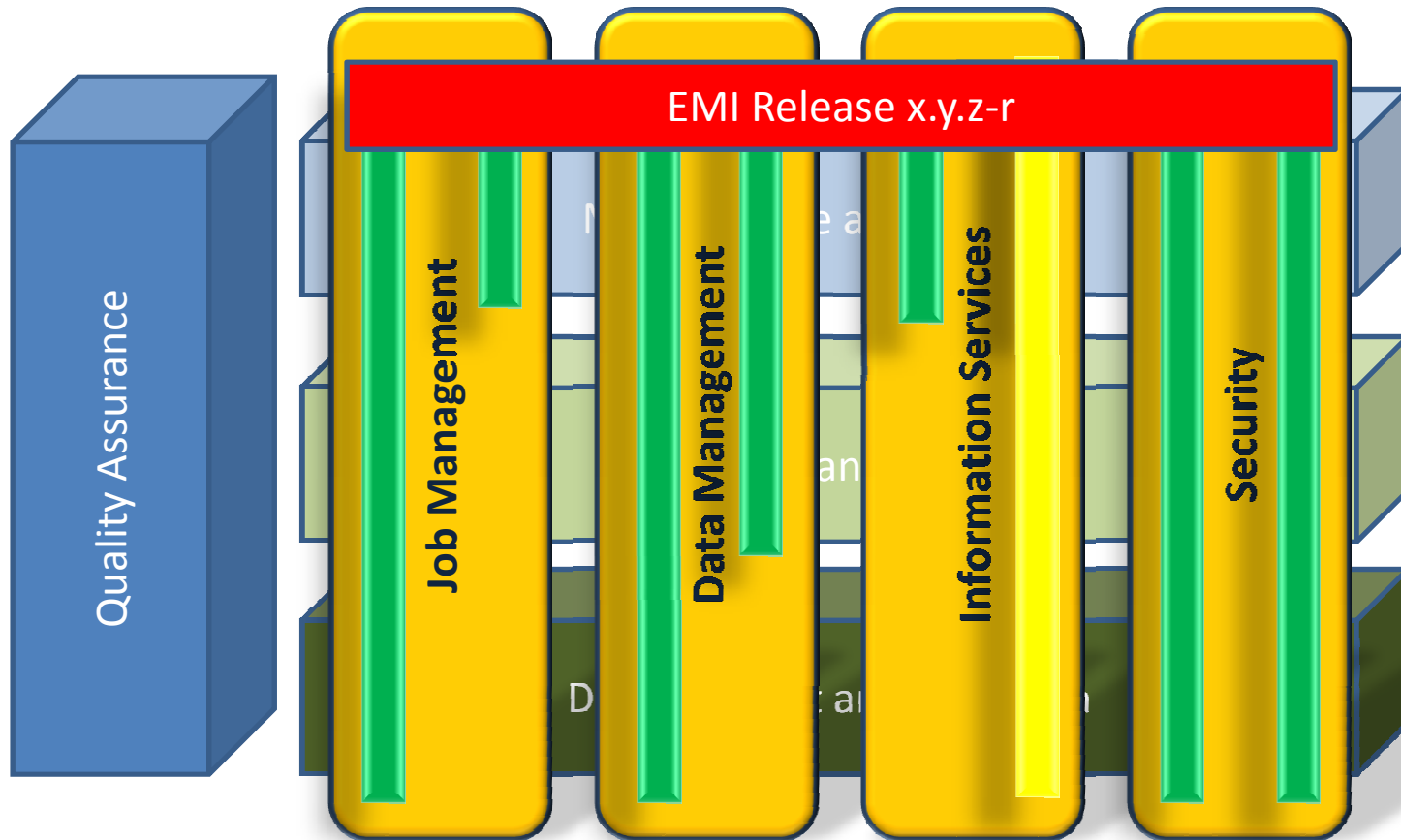


# Product Teams

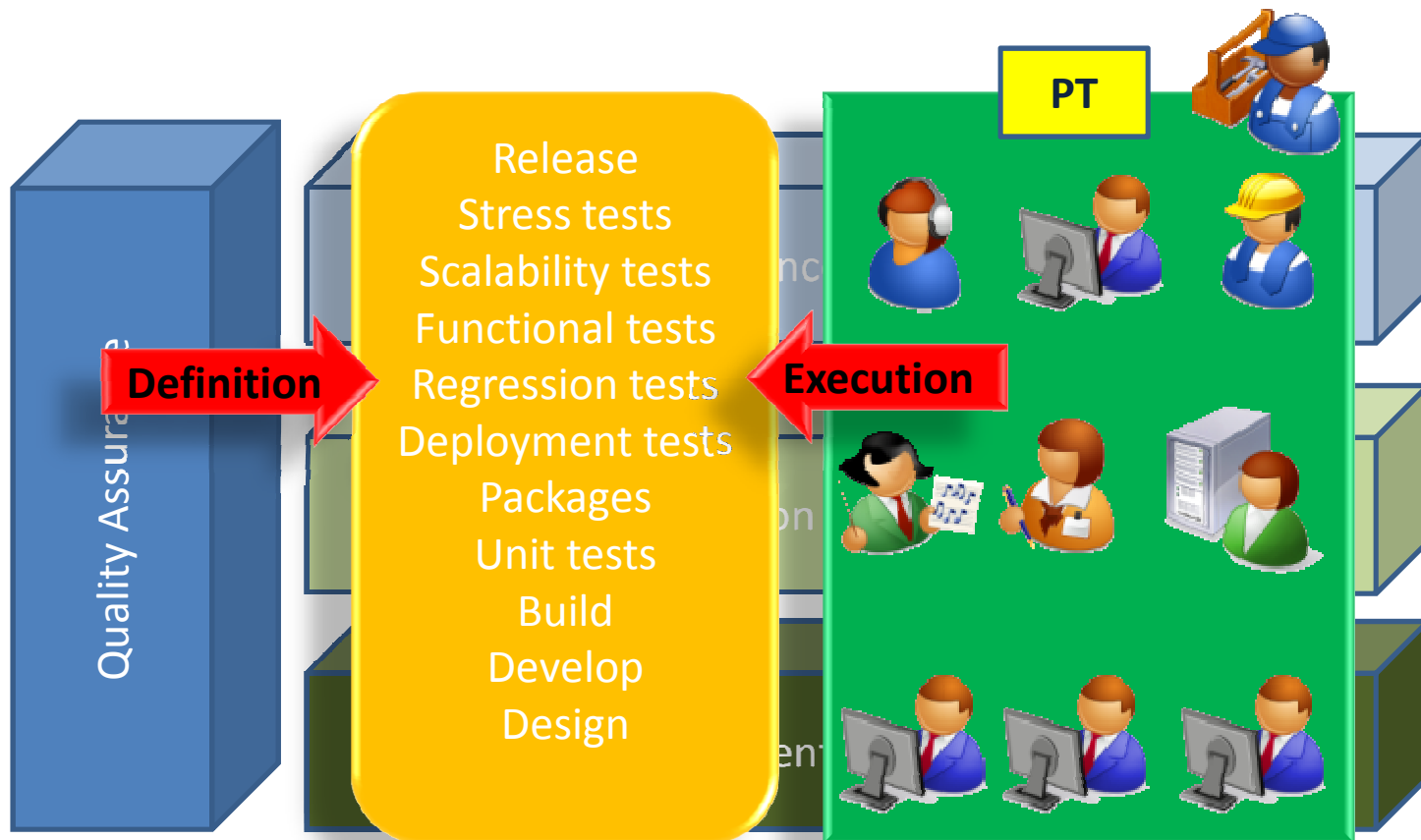
- Product teams are the services implementation teams within each area responsible to deliver software releases and all associated material.
- They perform the required technical tasks from design to release through implementation, testing, certification and 3<sup>rd</sup>-level support as part of one or more Work Package and according to what specified by the Project Technical Group and the QA policies.
- Product Teams are flexible, in the sense that they can be formed or closed as the corresponding products are introduced or obsoleted in EMI and allow adding or removing services as needed even from external contributors
- They provide a transparent and direct method to assign responsibility for a service



# Product Teams



# Software Engineering





# Tools: Build and Test System

- An ETICS-based service is provided as a build and test system
- If possible components should be built with ETICS
- If not possible, any other way is fine, but packages have to be registered in ETICS to allow other developers to access them and to perform tests
- All the automatable tests should be done with or accessible through ETICS
- The non-automatable tests need to be recorded in some way to be defined





# Tools: Bug Tracking System

- Existing bug tracking systems can be used
- EMI will have a specific bug tracking system to track high level issues
- If you (MW Distribution) need to change bug tracking system, this is a good opportunity to choose a good one for everybody and have it maintained in a single place



# Tools: Requirements Tracking System

- This is more difficult
- Is it done at the moment? How?
- In collaboration with EGI
- Needed to track agreements with EGI and other 'customers'
- Which release satisfies or WILL satisfy that requirement? (Planning)



# 3<sup>rd</sup>-Level Support

- EMI SA1 acts as 3<sup>rd</sup>-level user support
- Expected to use GGUS
- Not clear whether EMI SA1 is a single GGUS Team dispatching items to individual Product Teams or there is a GGUS Team for every PT (former is better)



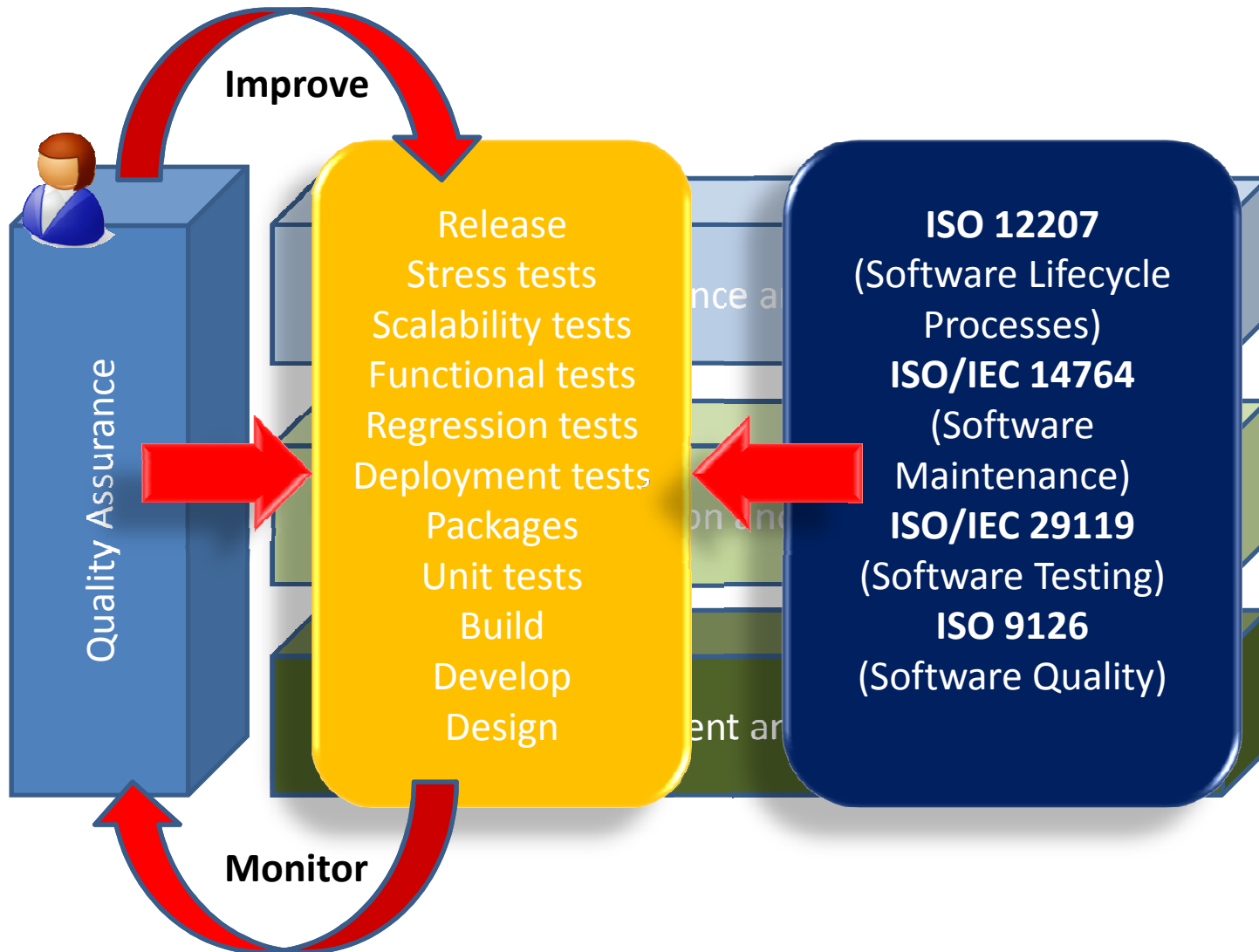


# 3<sup>rd</sup>-Level Support

- GGUS is the official entry point for users
- Users should be prevented or at least discouraged from submitting bugs directly in the bug tracking systems
- Bugs are filed by the PTs (within SA1) as the result of properly analysing a support request
- GGUS tickets are closed only when the problem is actually fixed by a new software release verified by the submitter



# Quality Assurance



# QA is not a constraint, it's a tool

- It is impossible to define a set of QA rules that can be applied all the times to all products
- It has to be flexible and useful
- But it has to be done
- Within a given set of rules, the actual thresholds have to be adjusted as necessary
- Ex:
  - new components are more buggy than old stable components
  - Old complex components cannot be given a full unit testsuite with 98% coverage overnight
  - New bug fixes can certainly come with a regression test as a rule, old fixes will be regression tested as practically possible
  - Try to make your tests automatable, you do the work only once and it can be reused by yourself and by others to verify their own components
  - Metrics are guidelines, not the law, use them as tools to spot weaknesses



# Basic Principles

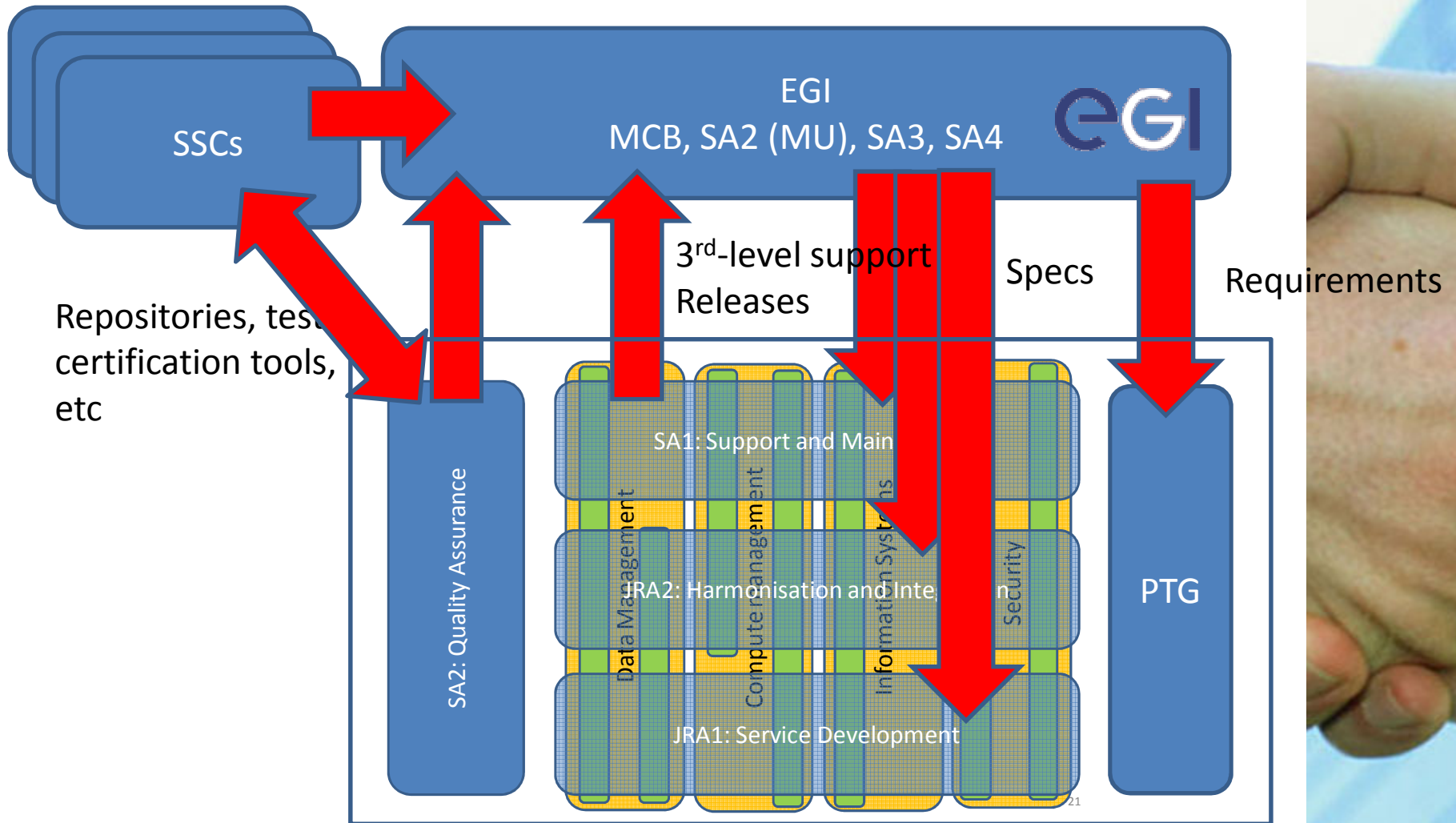
- Every PT is responsible to execute its own testing and certification according to the agreed QA rules
  - Necessary in a complex distributed environment
  - Central teams will never be able to certify everything in reasonable conditions and time
- All testing is done publicly and transparently, the tests and the test results are stored in the EMI test repository (ETICS)
  - Used to improve the process and the software
  - Use to monitor SLAs
- PTs using another PT products have a fixed grace period to validate new release candidates before they are released
  - Implicit ok = no bottlenecks
  - Implicit ok → is it enough?
  - Who is responsible in case of problems?
- If the agreed tests/criteria are not passed, the release is rejected (can happen at various stages)

# Releases and Release Policies

- **Major releases:** once or twice per year, may contain non-backward compatible changes
- **Minor releases:** a few times per year, fully backward compatible, may contain new functionality
- **Revisions:** every week or two weeks, only bug fixes
- **Emergency:** as needed, only specific bug fixes, use emergency release procedures



# User Communities





# Exploitation and Sustainability

- Collaboration with standardization bodies
  - Evolution of existing standards
  - Definition of new standards
- Inclusion of middleware in OS distributions
  - Packaging requirements, support, conflict management
- Service Level Agreements
  - Move towards a customer/provider model that could in time become a real commercial relationship





European Middleware Initiative

**Thank you**