

# Outsourcing Software Maintenance Services in a Cost Controlled Manner

James Eggleston

06/03/2015

© European Space Agency 2015. Published by The Aerospace Corporation with permission.

Issue/Revision: 1.0

Reference:

Status: Approved

ESA UNCLASSIFIED - Releasable to the Public



→ SERVING EUROPEAN  
COOPERATION AND INNOVATION

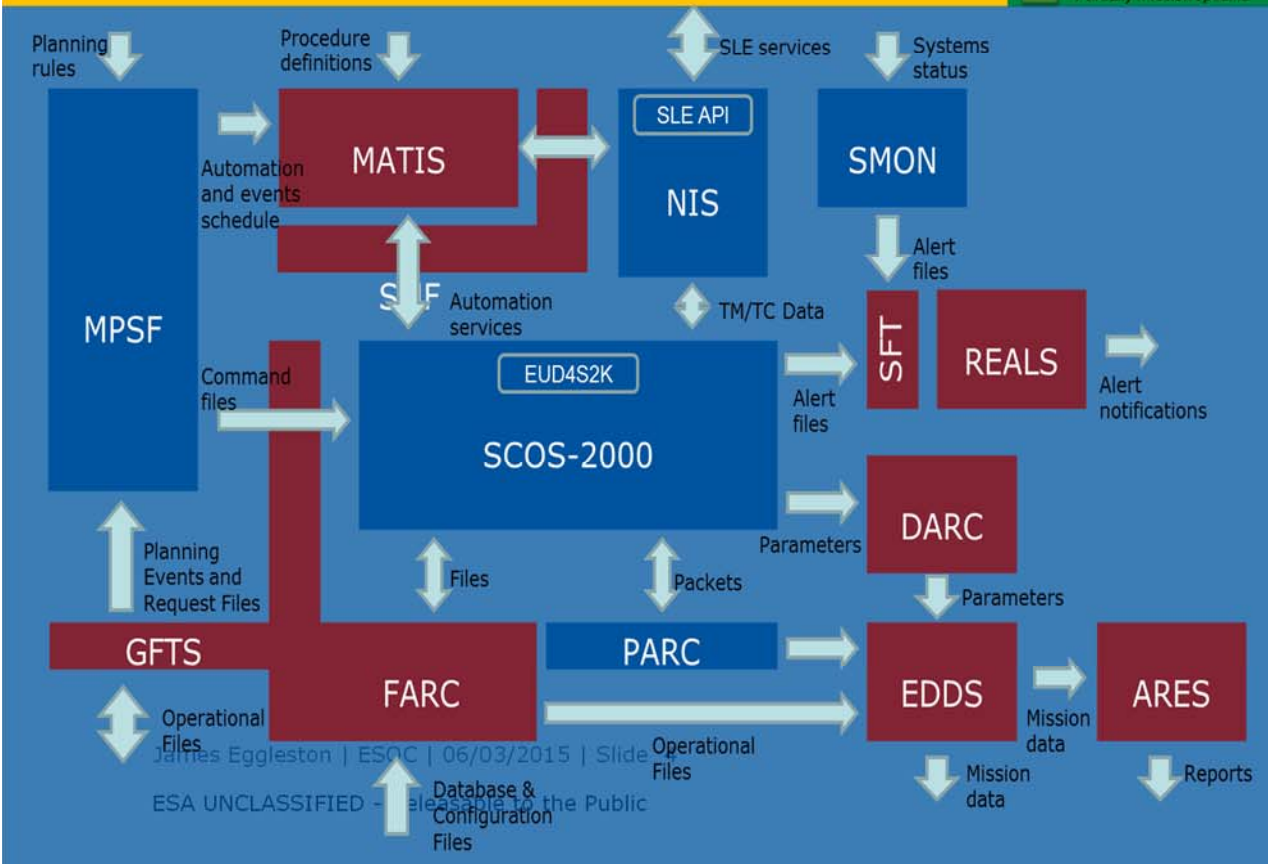
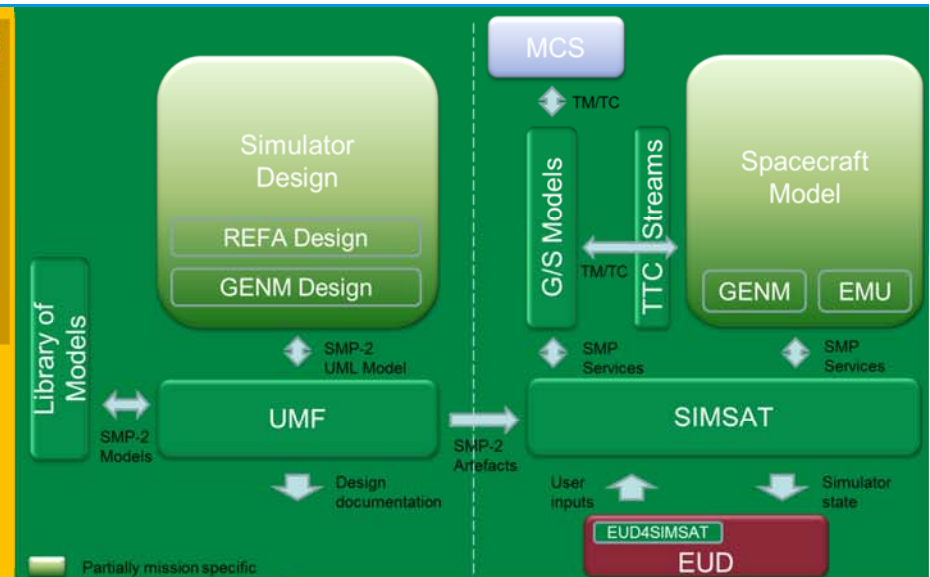
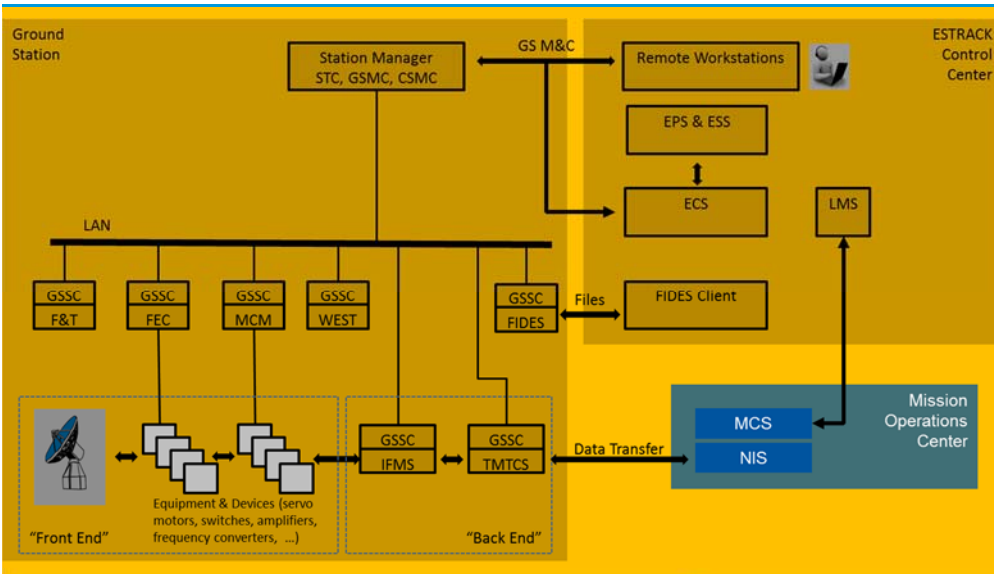
1. Background: ESA, ESOC, Infrastructure Maintenance
2. Maintenance Service: Concepts, Design, Implementation
  - a. Maintenance Policy, Standards & Processes, Charging Type & Concepts
  - b. Ticket Types, contract organisation & interfaces
  - c. Service Delivery: Service Dimensioning, Ticket Database, Reporting, Charging
3. Key Principles & Lessons Learned
4. Summary & Conclusions

# ESOC and it's Software Infrastructure



1. ESA / ESOC currently operating or is developing systems for operating approximately 25 missions.
  - a. Common software infrastructure is developed and maintained
2. ESOC infrastructure now contains ~35 software packages
  - a. Provides reusable libraries, frameworks and systems
  - b. Covers areas of mission control, mission planning, data distribution, ground stations and simulation
  - c. ~10 MSLOC, ~400 person years of effort
3. Maintenance of the infrastructure is a significant cost for ESOC
  - a. Constant pressure to justify, manage and reduce this cost
4. Previous maintenance approaches lead to company lock-in, poor visibility on cost drivers, quality issues, lack of contractor buy-in, and more than 10 independent maintenance contracts
  - a. Customer perception of expensive and uncontrolled maintenance





# Concepts: Maintenance Policy



The maintenance policy is based on the desire to...

1. Provide a cost-effective service to the customers, meaning
  - a. A standard, predictable and well-defined service to a fixed price
  - b. A rapid fix release in case of urgent problems
2. Promote knowledge sharing between the European companies
3. Keep the management costs down
4. Allow all companies in Europe to compete for the majority of our work, irrespectively of location
5. Get good value for money, whilst ensuring companies are treated fairly
6. Empowering efficient development and acceptance
7. Empowering efficient contract execution



# Design: Common, Standardised Process



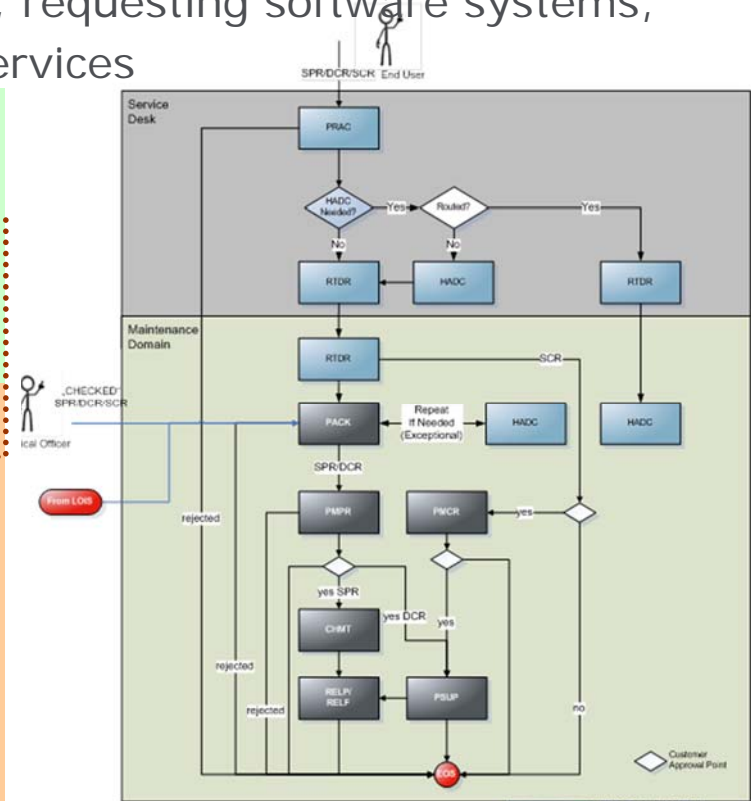
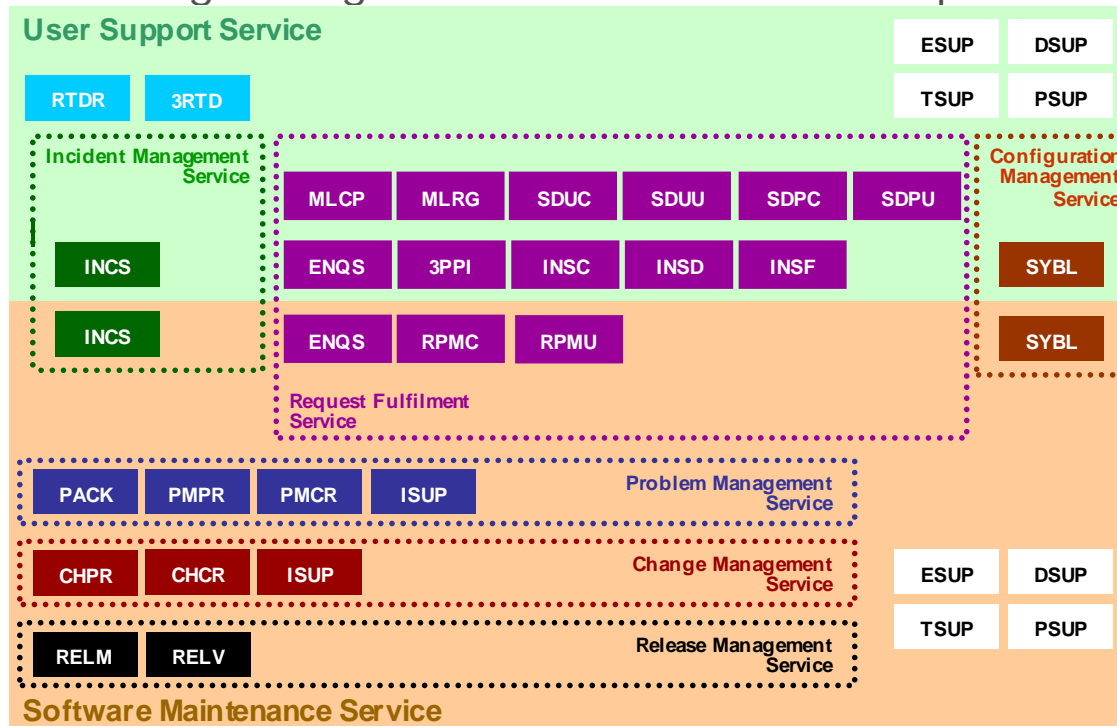
1. Each system follows a common baseline, set of technologies, development standards... hence we apply the same abstract process for maintenance and user services...
2. Maintenance Service follows "Information Technology Infrastructure Library v3" (ITIL3) standard for IT Service Management
  - a. "User Support Service" function and "Software Maintenance Service" function , containing a series of process "services", each defined by a given procedure
  - b. Together the processes provide all the necessary steps to provide the overall service
  - c. Service Level Agreement defines the expected response times for the various processes
  - d. An expected volume for each process for each system is provided in order to allow the correct team sizing
3. Each contractor bid provides quoted efforts for each process



# Implementation: Maintenance Processes Defined as Services



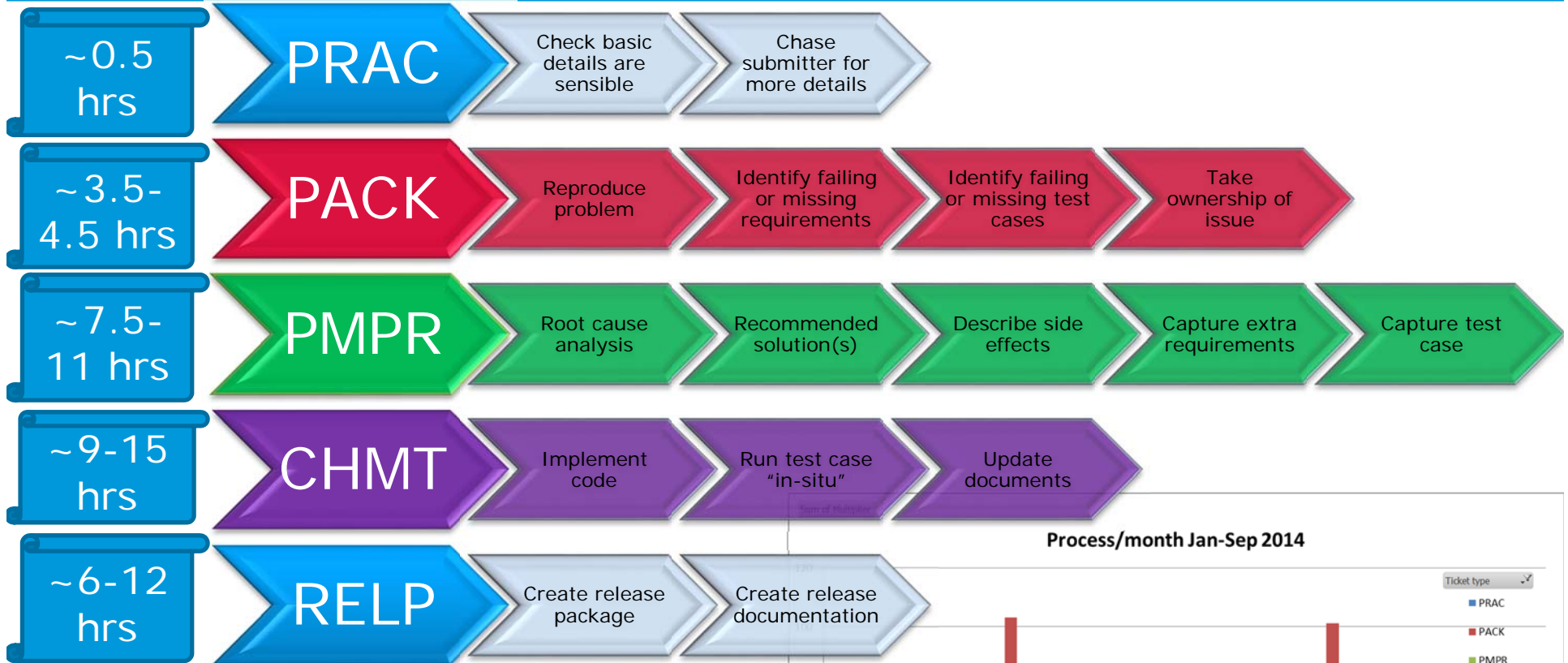
1. ~30 processes (inputs, steps, outputs), grouped into two types of contract:
  - a. User Support Service: answering questions, requesting software systems, granting access to software development services



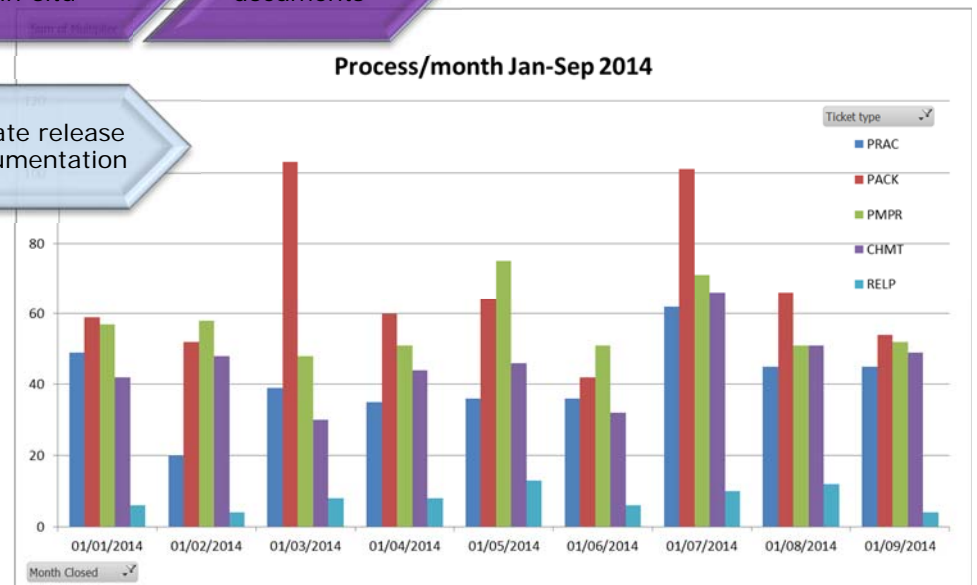
- b. Software Maintenance Service: acknowledgement of a problem report, change implementation of a problem report, software release



# Implementation Example: Software Problem Fix as a Service

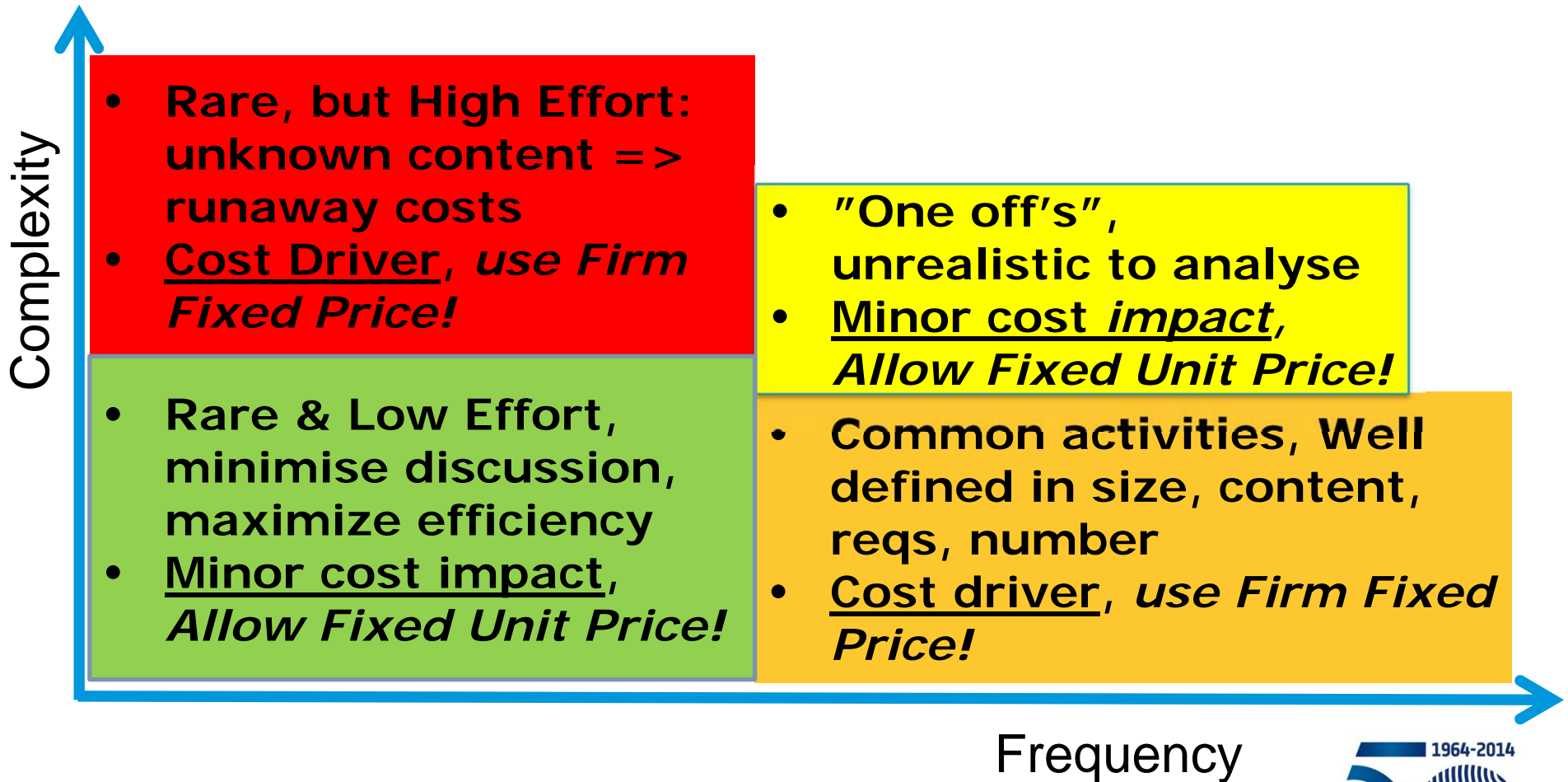


Process/month Jan-Sep 2014





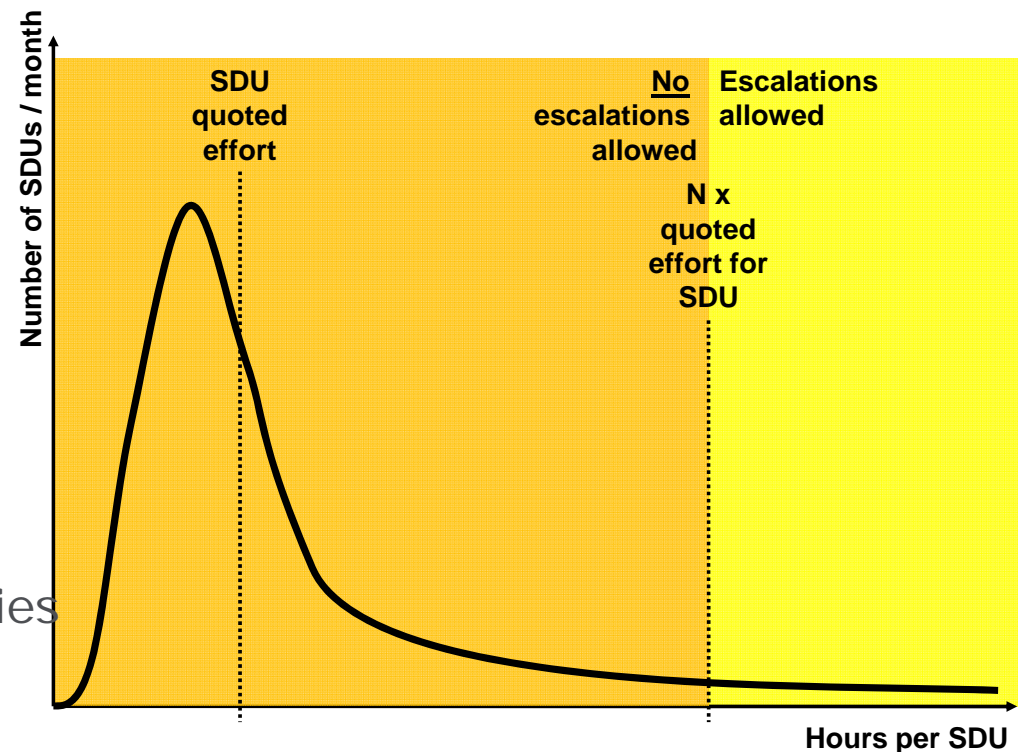
# Concepts: Being Effective and Efficient - Charging Types



# Design: Conversion between charging types



1. Allow Escalations from Firm Fixed Price model to the Fixed Unit Price model for unusual instances
  - a. Always with lightweight approval process, or via standing order...
2. Requesting one off Firm Fixed Price for (large) one off activities
  - a. Create "mini-proposal" for the specific task, with milestones, deliverables, etc



# Design: Describing the WP Content with Dimensioning



1. Dimensions define the quantity of service that we expect, and the distribution of across the processes in that service
2. Our processes are quantified as “Number of tickets per year”
  - a. This expected level can be used to calculate the expected cost at the end of any period
  - b. The dimensioning allows the contractor to size their team
3. The dimensioning can be adjusted each month, in line with service demand, expected activities, budget restrictions, etc
4. It has to be realistic...



# Implementation: Ticket Database & Reporting



1. Each process instance is recorded as a ticket in a ticket database
  - a. Each ticket represents a unit of work performed (service delivery)
  - b. Dedicated ticket lifecycle describes the states, attributes, transitions, and roles required
  - c. Database is a representation of all the activities of the maintenance service
    - Details of each service unit captured in the tickets....
2. Service monitored by database analysis - metrics like number of tickets by type or system, average ticket time, escalated tickets, etc.
3. Finances managed by adding contractual inputs, e.g. quoted ticket efforts & hourly rates, dimensioning, service levels, budget inputs.
  - a. Breakdown of costs (per system, per process, per user, per budget line, per company, per quarter)
  - b. SLA breakers , expected cost in any period, team size
  - c. Performance indicators



# Principles & Lessons Learned



1. Accept Change
  - a. All parties have to be willing to compromise. Hard contractual positions lead to inefficient or ineffective processes... doesn't mean lack of control, but change has to be accepted
  - b. We can change the service contents & rules, implies the contractor can change the price!
2. Atomic pricing
  - a. Each process "service" should be a single atomic action, e.g. analyse an issue, implement a fix, install a software package once.
  - b. Having 'open' process "services" leads to Fixed Unit Price like behaviour, destroys analysis
3. Minimal Administration and Discussion, both managerially and contractually
  - a. Managerial escalation is poisonous to efficiency - keep to the bare minimum
  - b. Sometimes it is better to not argue and just pay...



# Summary & Conclusion



1. ESOC defines a set of services to deliver Software Maintenance and Support Functions to ourselves and our users
  - a. Process "Services" are grouped together into contracts
  - b. Process "Services" are delivered either "Fixed Unit Price" like (for efficiency) or "Firm Fixed Price" like conditions (for cost drivers)
  - c. Process "Service" delivery is recorded by ticket - allows in-depth analysis
2. ~25,000 individual processes of ~30 types of service delivered in 3 ½ years covering ~35 software packages and ~20 support systems
3. Clear procedures for each process allow competition between companies for contracts, and eases companies working together
4. Application to all our systems creates a common vocabulary and an economy of scale
5. Services levels can be tweaked to respond to current or future needs

