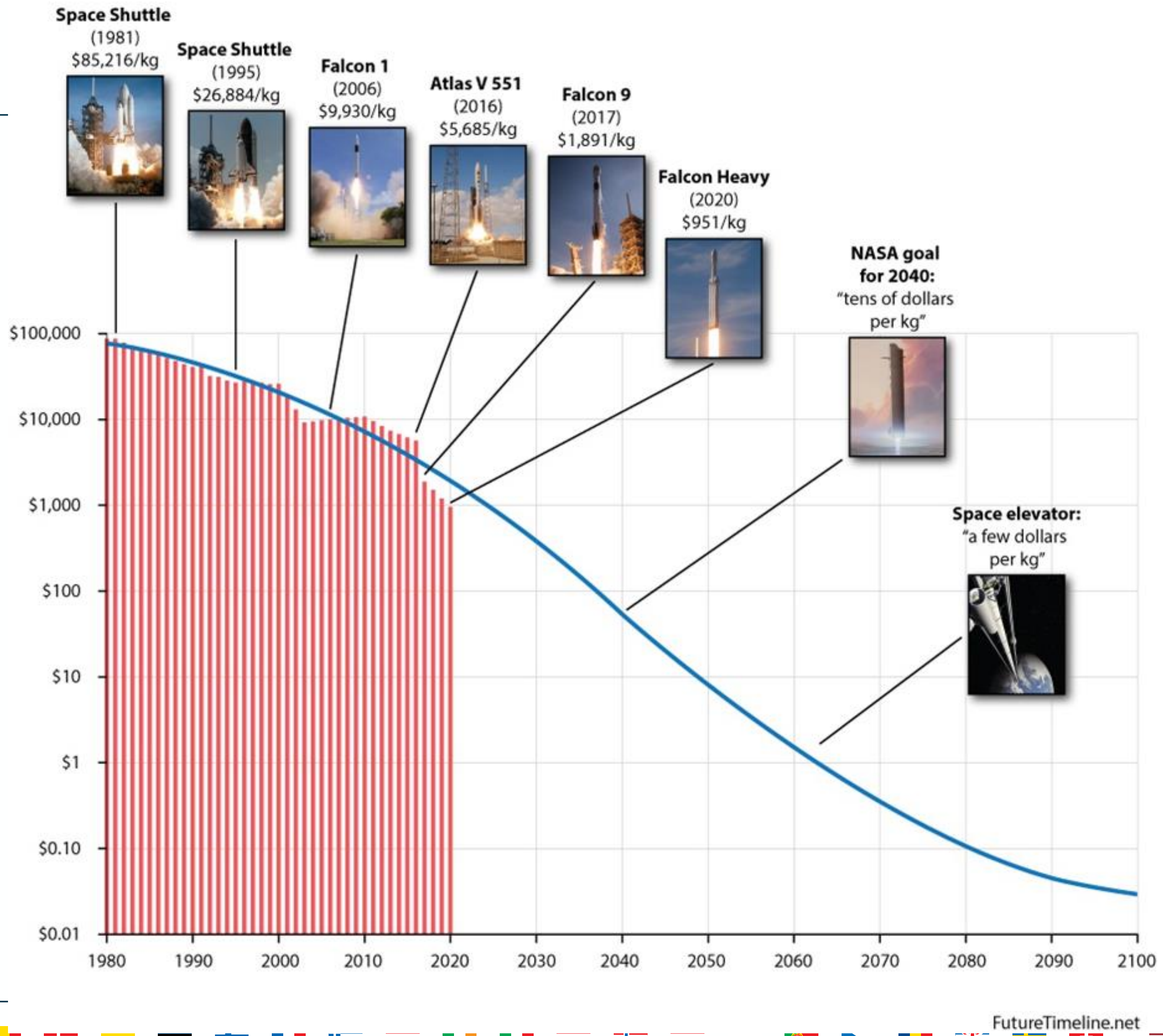
New Space paradigm implications for Software

David Escorial Rico (ESA/CSC-PLT)
Emanuele Gorini (LuxSpace)

Software Product Assurance Workshop 2023

27-Sep-2023



Cost per kilogram of payload delivered to low Earth orbit (logarithmic scale; inflation adjusted to year 2000)

From 80K+ USD in 1981 to 1K USD in 2020

What is New Space

Changes in the space industry in the last 10 years:

- Lower costs to access to space (10 times cheaper and decreasing)
- Increased miniaturisation of hardware at reduced costs
- Availability of resources (expertise, tools, standard components, funding…)
- Data-driven economy – demand for data – fast public access of data
- Economies of scale (from one-off to constellations of thousands)

Space is an environment where new markets can be developed

New Space impact

Impact on the space industry:

- New business models - direct access to large numbers of end users
- New sources of funding – entrepreneurs, private (VC) and institutional funding
- Changes in processes – quick response or first mover to markets
- Changes in products and technology – cost, risk and performance trade-offs – Commercial approach
- Stakeholders – institutions might provide funding, but they do not own or exploit space assets

<u>However, changes are not yet consolidated, still in transition</u>

- But what really defines a New Space company, approach, project, product..?
- New Space does not replace Traditional Space
- Much more than CubeSats

FOCUS on Software: Product, Processes … and Software Product Assurance

Software Product implications

Drivers

- Commercial input requirements
- Quick development (time to market)

Implications:

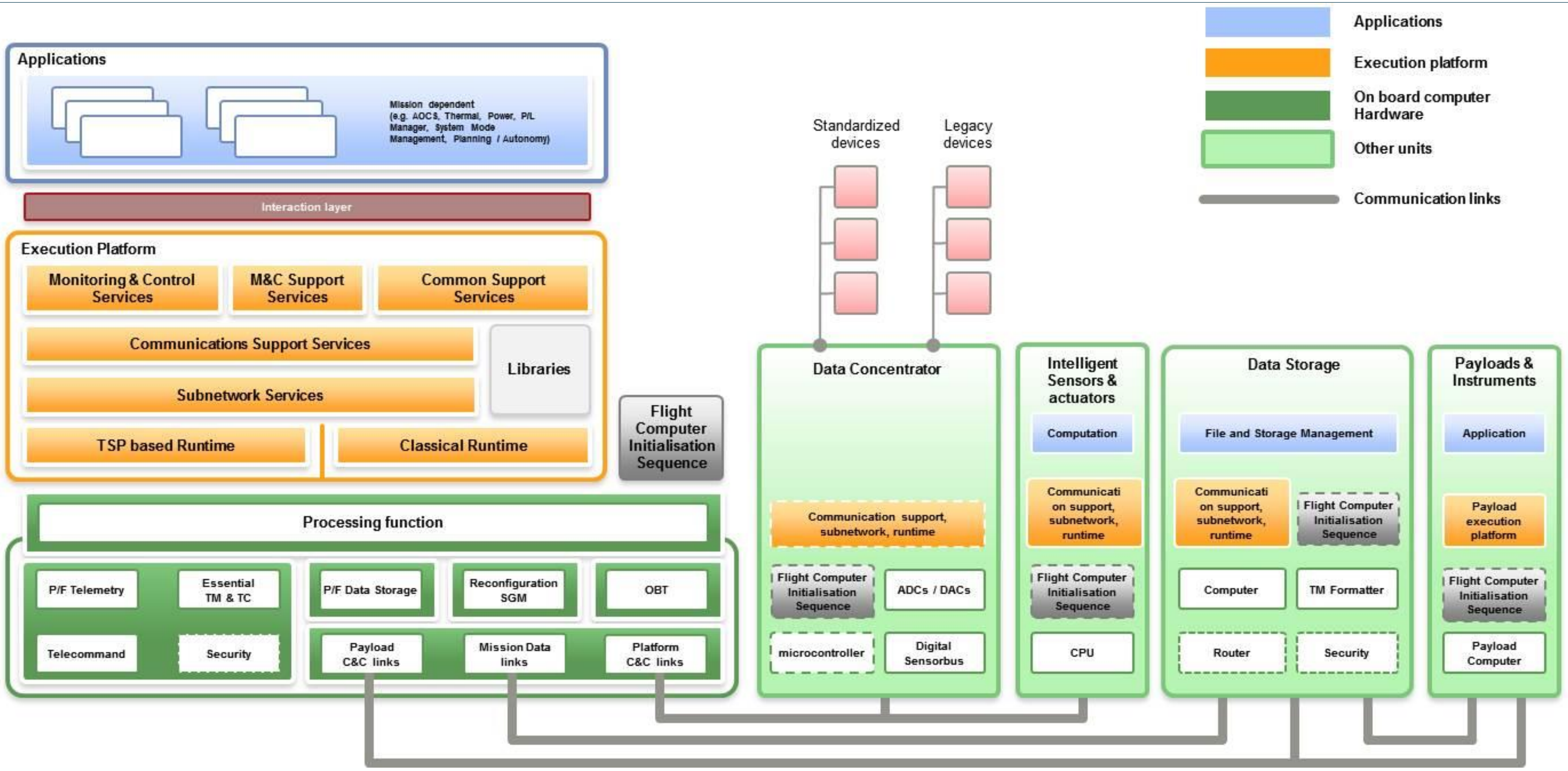
- Make use of resources already available, COTS/OSS (Open Source Software), development tools
- Some of the COTS are ‘de facto’ industry standards
- Performance and formal qualification are traded for cost and time
- But also: no need for specialised expertise (commercial processors and generic RTOS mean larger SW engineer pool available)
- Opportunities for software innovation (focus on value creation, e.g. AI)

Design for Reuse

Importance of Design for reuse

- SAVOIR (Space AVionics Open Interface aRchitecture) reference architecture concepts are relevant
- NEVER a single use Software architecture/product
- “Common core” product capabilities as technology base
- Missionisation: Scalability/Modularity
- Platform Software Developer Kits and SVFs for customers to develop their own software

SAVOIR Avionics System Reference Architecture



Software Process implications

Already established Software practices – in fact, New Space adopted them!

- Short iterations – Incremental software development lifecycles
- Modular development
- Minimum Viable Product deployment
- Fast upgrades with backwards compatibility
- High level of automatization (V&V in particular)

In combination with System development:

- Use of available evaluation boards for development, e.g. System-on-Chip
- Synchronised software product development and validation with evolutions of hardware (but HW is usually ‘longer’ iterations: redesign and manufacturing)
- Early integration of software in Flatsat
- Due to lower launch costs: in-orbit validations more frequent
- Radiation testing (HW COTS and SW mitigation mechanisms – established mitigation strategies)

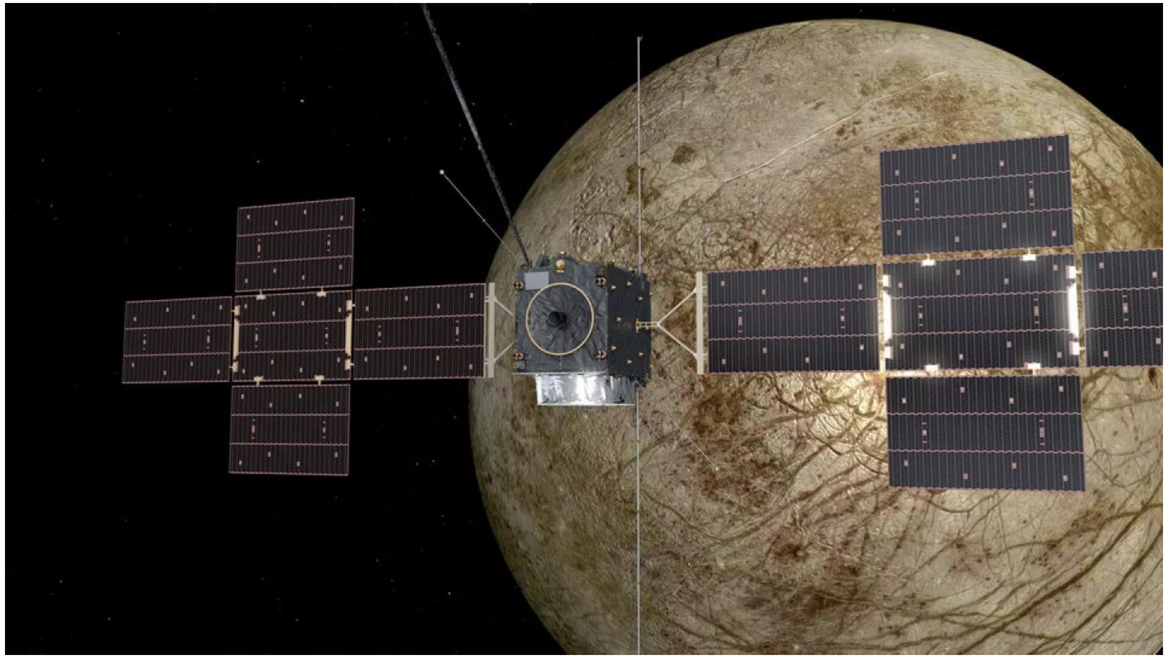
Software Quality and New Space

Clash of principles?

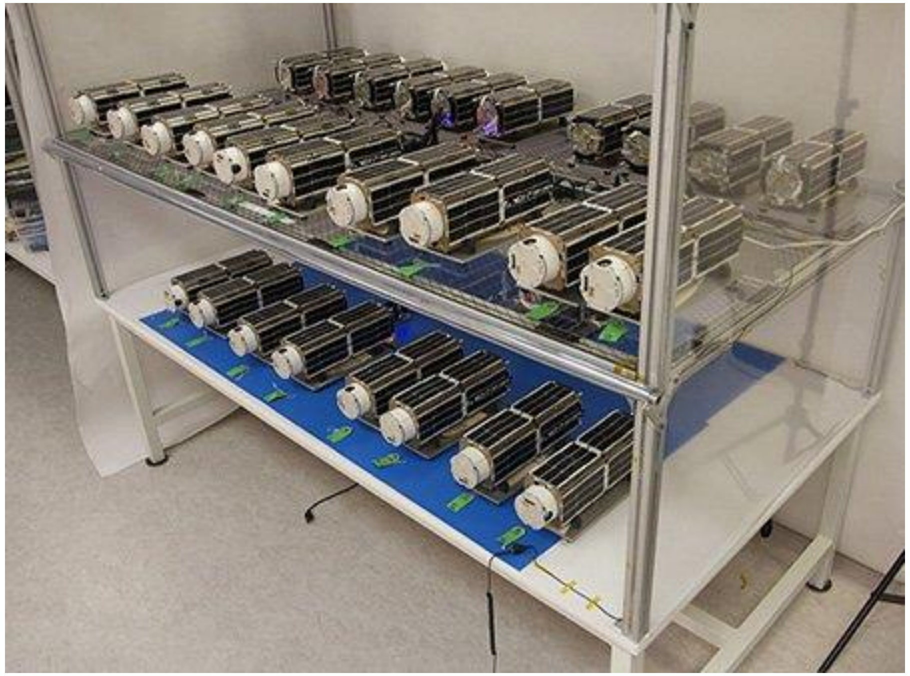
- “Fail Fast, Fail Often” vs “Right the First Time”
- Failure CAN BE an option (* in some cases!) – accept more risk (traded for less cost and time)

Quality as ‘Fit for purpose’ and ‘Value for money’

- Meeting the expectations of the stakeholders (customers, users, funders…), focus on providing value



JUICE -An artist's impression of Juice flying by Europa - Copyright ESA



Dove constellation 3m - Copyright Planet Labs Inc

Software, ECSS and New Space

But what about ECSS?

- Formal requirements from Institutions (e.g. Agencies) reduced to safety and environmental factors (e.g. space debris)
- Quality can be assured using standards (correctly), but quality* (as defined before) can also be achieved without them
- They collect useful practices and guidance – e.g. Design for Reuse, COTS management, SAVOIR,…
- Open new markets (institutional, critical missions,…)

But…

- Upfront costs (learning curve, process establishment)

Working in the right direction – the 'good old tailoring':

- ECSS 4.0 core requirements
- ESA mission classification

But still, it is an Institutional-oriented solution (milestone reviews, documentation, …)

Software PA and New Space – approaches

Some ideas:

- Software Core capabilities SHALL be reliable (specially in constellations) – products and services are built on top of it
- Focus on core assurance objectives of Software development processes, and automatize them
- From compliance/non-compliance of product/process requirements to wider set of measures and qualitative considerations
- Software product maturity and quality are built iteratively
- Simple step by step approach:
    1. Being aware of the process – how do we develop software?
    2. What are our software key processes (e.g. testing or deployment)?
    3. How can we improve those processes?
    4. What are the basic measures we can collect (e.g. coverage)?
    5. When and how to report PA and Verification results to stakeholders (e.g. go/nogo assessment)?