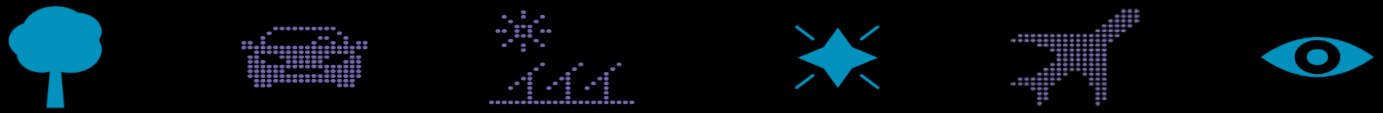


Practical application of SPARK:

a business case and roadmap for new users

June 2018

ALTRAN
Intelligent Systems.

Overview

ALTRAN Group

+44,000
EMPLOYEES

~€3Bn
2017 REVENUES

20+
COUNTRIES

- **ALTRAN…**
- Is a **global leader in Engineering and R&D** Services
- Works with 300 of the top 500 Companies in the world
- Is active in many **different industries**, allowing us to have second to none insights and cross industry fertilization capabilities.
- Is the 1st engineering partner of Airbus, PSA, and ranked as Strategic partner by more than 50 companies…

ALTRAN
Intelligent Systems.

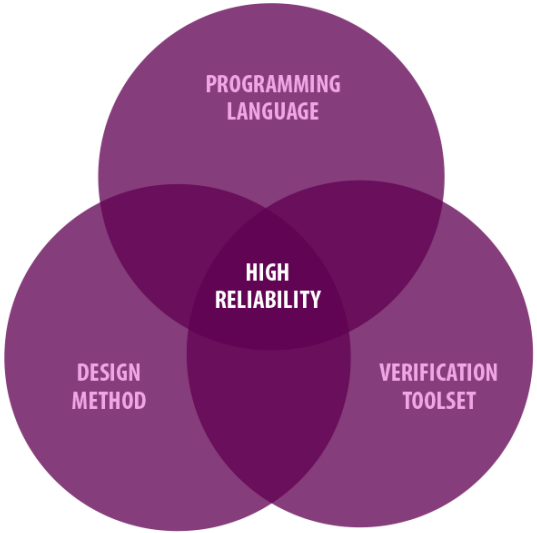
Intelligent Systems Expertise Centre – Our World:

ALTRAN
Intelligent Systems.

What is SPARK?

- SPARK is …
  - › a language
  - › a toolset
  - › a design approach

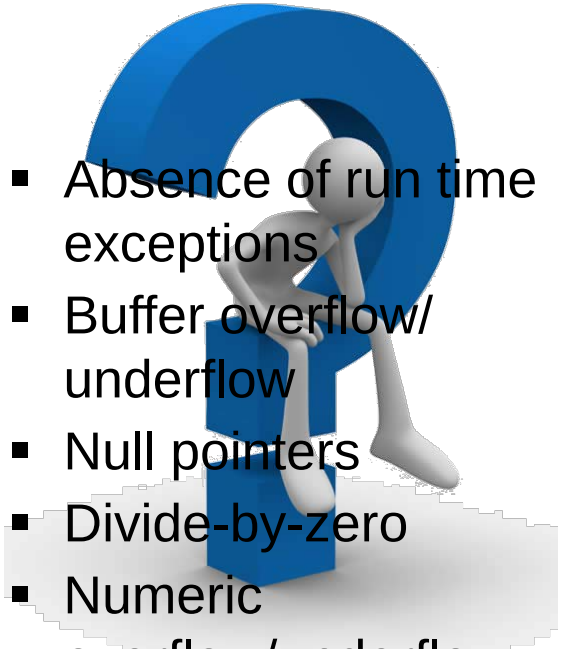
  … for the development of high-integrity software



- And a way to (formally) address a rich set of verification objectives …

Which Verification Objectives?

- Language subset
- Coding standard
- Variable initialisation
- Aliasing
- Data flow
- Information flow
- Type safety



- Absence of run time exceptions
- Buffer overflow/ underflow
- Null pointers
- Divide-by-zero
- Numeric overflow/underflow

- Contracts
- Security properties
- Safety properties
- Functional correctness
- ...

Context Matters

- No single answer to the above. It depends on:
  - Integrity level
  - Regulatory framework
  - Overall assurance plan
  - Assumptions
  - Dependencies
  - Where to draw the boundary

… to name a few

Levels of Assurance

- For new users, a framework has been proposed by AdaCore & Thales [1] that breaks the verification objects into a scale of “SPARK Assurance Levels”:

1. Stone level - valid SPARK
2. Bronze level - initialization and correct data flow
3. Silver level - absence of run-time errors (AoRTE)
4. Gold level - proof of key properties
5. Platinum level - full functional correctness

- Successfully applied by Thales [2]

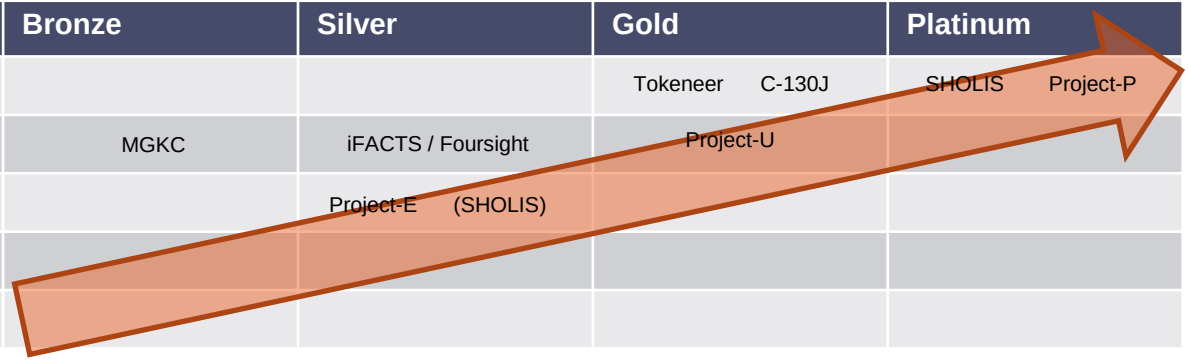
Historical Perspective

Software Integrity Level		SPARK Assurance Level			
DAL	SIL	Bronze	Silver	Gold	Platinum
A	4				
B	3				
C	2				
D	1				
E	0				

- Stone level is not represented as it is more an intermediate level during adoption of SPARK than a target assurance level.
- “SIL-0” is an informal (but widely-used) term => “software below SIL-1 but which is still well-engineered”
- Other scales are also relevant for secure systems eg. Common Criteria

Previous Projects

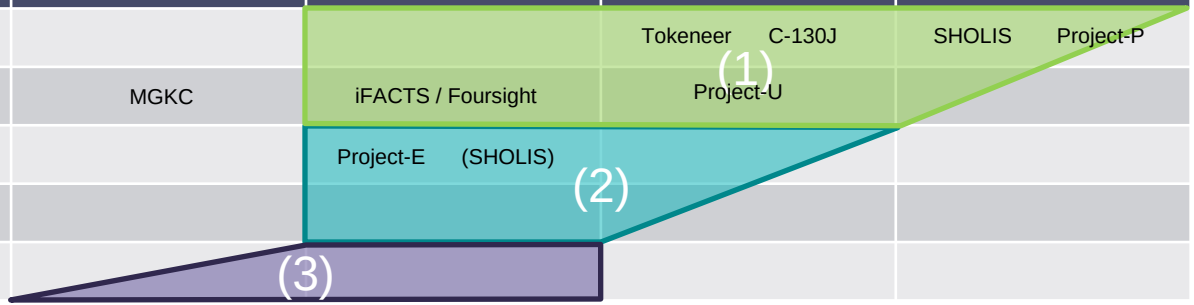
Software Integrity Level		SPARK Assurance Level			
DAL	**SIL**	**Bronze**	**Silver**	**Gold**	**Platinum**
A	4			Tokeneer C-130J	SHOLIS Project-P
B	3	MGKC	iFACTS / Foursight	Project-U	
C	2		Project-E (SHOLIS)		
D	1				
E	0				



- Previous Projects:
  - Tokeneer, C-130J, SHOLIS (SIL-4 subset), Project-P
  - MGKC, iFACTS/Foursight, Project-U
  - Project-E, SHOLIS (SIL-2 subset)
- Underlying trend higher SIL => higher assurance level

Guidance

Software Integrity Level		SPARK Assurance Level			
DAL	SIL	Bronze	Silver	Gold	Platinum
A	4			Tokeneer C-130J	SHOLIS Project-P
B	3	MGKC	iFACTS / Foursight	Project-U	
C	2		Project-E (SHOLIS)		
D	1				
E	0				



- We identify three broad categories for guidance:
  1. At highest SIL/DAL, Silver is a minimum and may go up to Platinum
  2. At medium SIL/DAL, Silver is a minimum and could go up to Gold
  3. At lowest SIL/DAL, Silver is still default but could be weakened to Bronze
- Silver is the “Gold Standard” ☺

The SPARK Boundary

- An equally important decision (to Assurance Level) is where to draw the boundary
- SPARK even allows software to be safely partitioned within the same application
- SHOLIS is an example of this:
  - SIL-4 part at Platinum (Full functional proof)
  - SIL-2 part at Silver (AoRTE)
- The non-interference between different sections of the code was assured by the use of information flow analysis
- Contracts (“Derives”) – not considered by the Assurance Levels – are attached to each subprogram and checked by the tools

Hybrid Approaches

- Project-P is most recent, and takes advantage of the dual nature of contracts in SPARK 2014
- A hybrid verification strategy (ConTestor) where the SPARK contracts are being used for dual purpose
  - Static formal verification (proof) of implementation against contracts
  - To provide an oracle (expected outcome) on dynamic tests
- Test cases for the integrated code are generated using constrained-random test generation
- If no exceptions are raised during execution then the code passed the test case
- Completeness is measured in terms of a set of independently specified verification conditions

SPARK – The Business Case

- We don’t just do this for the ‘normal engineering reasons’ (time/cost/quality) – for which there is plenty of evidence eg. C-130J [3], SHOLIS [4]

- It also gives us commercial differentiation through the ability to offer software warranties …

Legal Context (UK & Europe)

- Products are covered by a body of law and can be guaranteed

- Software is not a “product”
  - Buyer is not protected by product law
  - Best efforts are good enough
  - (Aside: Installing it onto hardware makes it a product!)

- Warranties help the buyer by shifting some of the risk/responsibility onto the supplier

Warranties

- Altran believe SPARK/Correctness-by-Construction is better & cheaper
- How do we share this benefit?..
- One way is to use a warranty
- Both parties need to buy-in to get the benefit: an agreed specification
- Warranties give the customer assurance on what they are buying - guaranteed quality from day one
- This gives us a USP: as far as we know our warranties are unique in the world of bespoke software

Warranty Terms & History

- Typically 3-5 years
- Covers fixing software but not consequential costs
  - Because these are not under our control
- Three levels of fault types vs. service levels (eg. low->include in next build)
- There is no discount for not having the warranty: our development processes are tried and trusted!
- Warranties on majority of previous software projects
- Warranty claims? – Yes: 3 in 20+ years (of which one cosmetic)

Conclusions

- SPARK is a powerful verification toolbox, but it can be daunting to new adopters
- The SPARK Assurance Levels framework helps new users navigate through the choices – as demonstrated by Thales
- We have validated the framework against 20+ year history of application by Altran & other users
- Engineering benefits of SPARK are well documented
- The hidden benefit for a service provider such as Altran is the commercial differentiation it gives us, manifest through software warranties

References

- [1] Implementation guidance for the adoption of SPARK, AdaCore & Thales, Release 1.0 https://www.adacore.com/books/implementation-guidance-spark
- [2] Climbing the Software Assurance Ladder - Practical Formal Verification for Reliable Software, Dross et al., AVoCS 2018
- [3] Martin Croxford and James Sutton. Breaking through the V & V bottleneck. In Ada Europe 1995, volume 1031. LNCS, 1996.
- [4] Steve King, Jonathan Hammond, Roderick Chapman, and Andy Pryor. Is proof more cost-effective than testing? Software Engineering, IEEE Transactions on, 26(8):675–686, 2000.

ALTRAN

Intelligent Systems.