intecs
the Brainware company

AUTOMOTIVE SPIN ITALIA

10° Workshop on Automotive Software & Systems

Milano – 25 Ottobre 2012

Hotel Four Points by Sheraton, via Cardano,1 (zona Stazione Centrale)

Automotive SPICE e strategie efficaci di V&V in contesti progettuali medio-piccoli

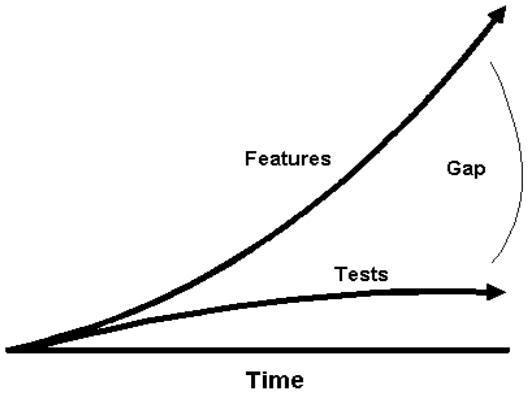
fabio.falcini@intecs.it

SPI Consultant and iNTACS Principal Assessor (Intecs spa)

What is about?

This presentation is an excursus along V&V in the context of ECU SW Projects in the "small settings" arena.

- ***The processes related to V&V are the ones definitively more challenging for such a kind of organizations.***



The aim is at presenting a set of V&V approaches as a valuable trade-off between Automotive SPICE requirements and a sustainable and effective V&V implementation for an average "small setting" project.

What is NOT about?

Definition of V&V Strategies intrinsically compliant with Automotive SPICE.

- ***Strategies needs to be adequate to the context!***
- ***Strategies needs to be well detailed and deployed correctly and systematically!***

Introduction

The ideas behind this presentation emerge from author working experiences at Suppliers of small-to-medium ECUs.

- ***Focus on ECUs based on 8 bit and 16 bit microntrollers***
- ***Software is traditionally hand-written (i.e. no Model Based development)***

In "Small Settings" there are well-known reasons why Automotive SPICE is so hard to be applied, above all:

- ***High Costs of establishing the necessary infrastructure***
- ***Large number of roles which must be filled by a restricted number of staff***
- ***Amount of information that must be assimilated to properly interpret the Automotive SPICE model***

The "Small Setting" Context

OEMs typically require ECU suppliers to apply the same SW Standards regardless the size of the project and of the ECU Supplier.

What are these standards typically?

- ***MISRA C 2004***
- ***<u>Automotive SPICE</u>***
- ***ISO 26262 (if functional safety comes into picture)***

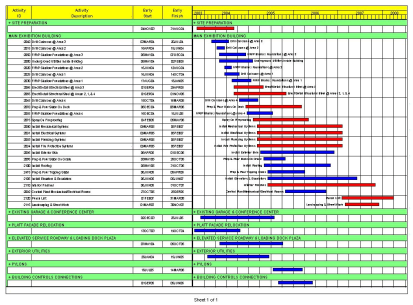
.... and it sometimes does!

The “Small Setting” Context (2)

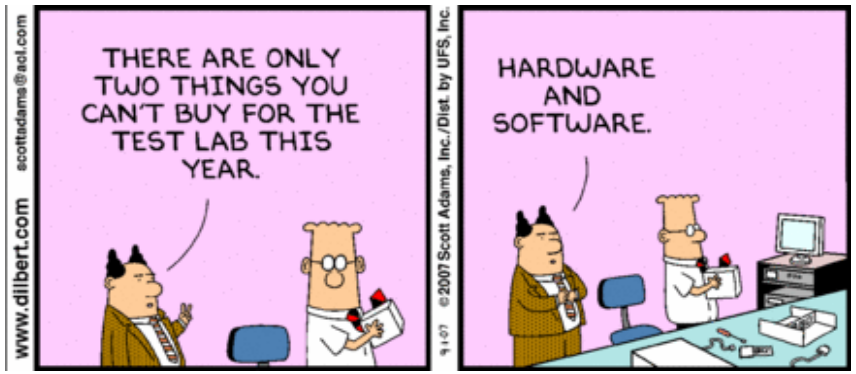
... and since we are in Automotive:

- ***Schedules are always compressed***

<u>and test windows get squeezed!!!</u>



- ***Not much room for investments (man-power, tools, automation ...)***

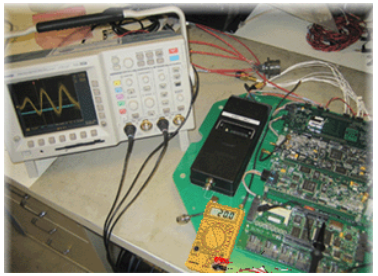


The “Small Setting” Context (3)

What is a typical scenario in a “small setting” project?

1 or 2 SW Engineers developing the project

They typically develop and test/debug the SW on Target and Emulator.



1 System Tester that tests the whole system

He/She tests heavily focuses on Customer requirements and applicable norms.

V&V in Automotive SPICE

Automotive SPICE Engineering Process Areas with V&V relevant content:

- ***ENG.6 – SW Costruction***
- ***ENG.7 - SW Integration Testing***
- ***ENG.8 - SW Testing***
- ***ENG.9 - System Integration Testing***
- ***ENG.10 - System Testing***

www.automotivespice.com

Filling the Gap

Typical result from an Automotive SPICE Gap Analysis in a generic Small Setting on an average project (Not "Best in class").

intecs informatica e tecnologia del software
Brainware Company

3

GAP ANALYSIS RATING

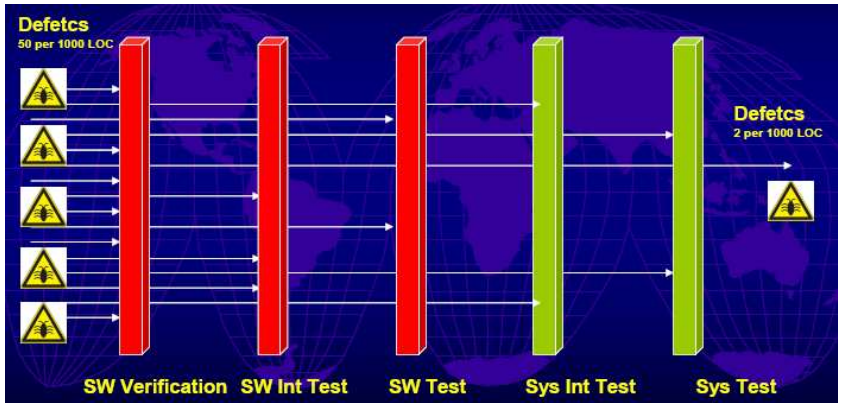
rating			
N	P	L	F

ID	Process name	Capability	Implementation Level (L1)	Major Criticality
MAN.03	Project Management	1	60%	Definition, planning of SW activities is poor. Accordingly monitoring is reactive.
SUP.01	Quality Assurance	0	50%	Very low Focus on SW development activities
SUP.08	Configuration Management	0	50%	This process area is not supported adequately in terms of tools.
SUP.09	Problem Resolution Management	1	60%	The activities of these processes are not tracked adequately.
SUP.10	Change Request Management	1	60%	
ENG.02	System Requirements Analysis	1	60%	This process is not systematic and needs strong improvement and tuning (e.g. consolidated list of the system features) .
ENG.03	System Architectural Design	1	60%	Some aspects of this process area are mingled with ENG.2.
ENG.04	Software Requirements Analysis	0	50%	The documentation (SW Requirements) is often NOT up-to date and is not an adequate support for the ENG.8.
ENG.05	Software Design	0	50%	Some aspects of this process area are mingled with ENG.4 (SW Description Document). Some important design choices are not documented.
ENG.06	Software Construction	0	50%	SW Verification is conducted effectively but informally and very partially.
ENG.07	Software Integration Testing	0	50%	Formality is very low.
ENG.08	Software Testing	1	60%	Formality is low. Traceability to SW Requirements is poor.
ENG.09	System Integration Testing	0	50%	Formality is very low.
ENG.10	System Testing	1	70%	Formality is low. Traceability to System Requirements is poor
ACQ.04	Supplier Monitoring	Not Applicable		

A lot of stuff to do in order to improve!

So What Strategies and Techniques?

- ***SW Units Verification***
- ***SW Integration Testing***
- ***SW Testing***
- ***System Integration Testing***
- ***System Testing***



First of All

Setup a light-weight and usable Traceability Infrastructure:

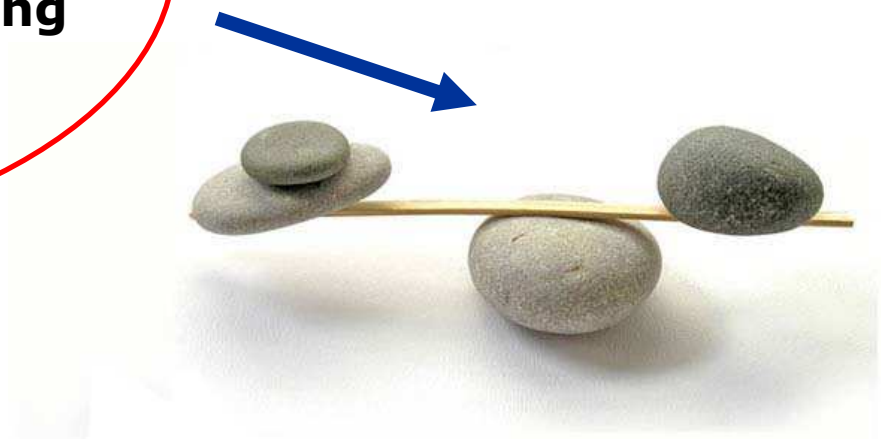
Microsoft Excel - W5-PL7 TM

Update RTM with Req. IDs

	SW Requirements (Baseline. 00.F)		TRACEABILITY			
	Req. ID	Description	Sys Requirements	Sys Arch	SW Arch	SW Test Case
4	SW-001044	Frequency outside valid ranges	SYS-001	uC_SNS_KL 30	CMDMGR	TST-000146 - A
5	SW-001046	Operative modes definition	SYS-002	uC_SNS_VM OT	CMDMGR	TST-000052 - A TST-000054 - A TST-000056 - A TST-000057 - A TST-000058 - A TST-000059 - A TST-000060 - A TST-000063 - A TST-000064 - A TST-000066 - A TST-000067 - A
6	SW-001189	Plausibility check	SYS-003	uC_SNS_VS 1	CMDMGR COM	TST-000051 - A
7	SW-001223	Target voltage with TV = 0% and 188%	SYS-004	uC_SNS_TM P	CMDMGR	TST-000052 - A
8	SW-001190	Sleep mode behaviour	SYS-005	uC_SNS_TM P	VREG STMGR	TST-000054 - A
9	SW-001049	Shut-down execution	SYS-005	uC_SNS_CU RR	STMGR	TST-000055 - A
10	SW-001050	Wake-up execution	SYS-006	uC_CMD	IO	TST-000056 - A
11	SW-001051	Command law	SYS-006	uC_PWM	CMDMGR	TST-000057 - A
12	SW-001052	Overdrive	SYS-007	uC_DIAG	CMDMGR	TST-000058 - A
13	SW-001191	Void	SYS-008	uC_MAIN_E N	CMDMGR	TST-000059 - A
14	SW-001054	Handshake notification	SYS-009	uC_IRQ	CMDMGR DIAG	TST-000060 - A
15	SW-001055	Beginning of handshake notification	SYS-010	uC_MAIN_EN	CMDMGR COM	TST-000061 - A
		Acknowledge of a new handshake	SYS-011	uC_DIAG	CMDMGR	TST-000062 - A

Focus on Balanced Strategies

- **SW Units Verification**
- **SW Integration Testing**
- **SW Testing**
- **System Integration Testing**
- **System Testing**



It is strongly advisble to devise jointly the overall V&V strategy to optimize the effort and to compensate weaknesses of each V&V layer!

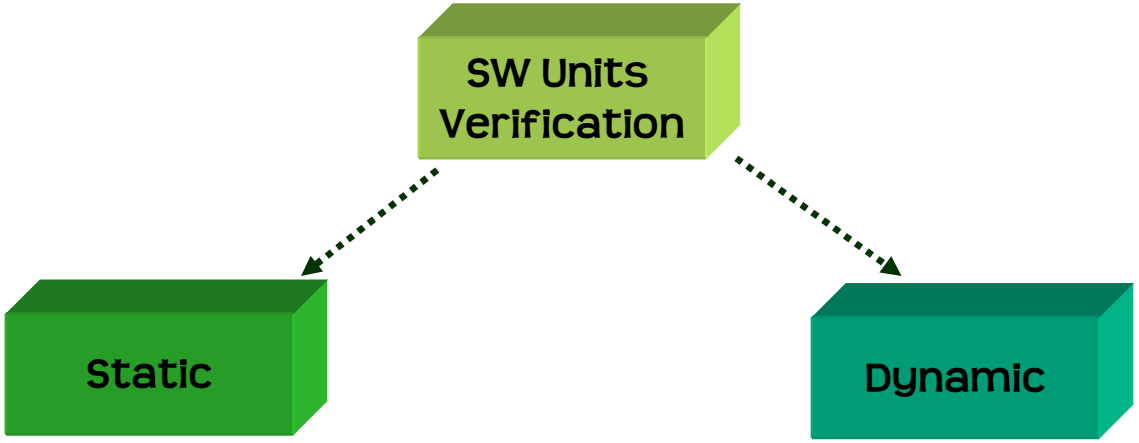
Aggregate Activities not Strategies

When V&V strategies and objectives are clear, activities can be conveniently and effectively combined.

A typical opportunity is:

- ***SW Integration Testing***
- ***SW Testing***

ENG.6 – SW Units Verification



Unit Testing is made difficult by:

- ***HW constraints do not encourage unit test on host (8bit, 16bit, flash)***
- ***Very time consuming***
- ***Poor documentation at SW Design level***

ENG.6 – SW Unit Verification (2)

Full speed ahead on:

- ***Functional Code Inspection***
- ***Enforcement of defined Corporate Coding Standard***
- ***Respect of well selected SW Metrics***
- ***Respect of MISRA C 2004 Rules***

Static Analysis techniques play a key role, especially in increasing small size software quality!

ENG.6 – SW Unit Verification (3)

Functional Code Inspection

- ***Functional Code Inspections***

Pair programming

- ***Driver and Navigator working together on one task***
- ***Roles changing often***
- ***Collective responsibility for outcome***
- ***Bringing together of multiple perspectives, experiences, abilities, and expertise***

ENG.6 – SW Units Verification (4)

Effective UT approaches may include:

- ***Perfoming Unit Testing of selected critical functions on emulator depending on the application and its context***
- ***Performing Unit Testing of selected functions algorithmic intensive on Host***
- ***Performing Unit Testing of selected single lines of code (LOC) – typically complex mathematical statements – on Host***
- ***Application of advanced techniques (e.g. pairwise testing)***

ENG.6 – SW Units Verification (5)

Pairwise Testing:

- ***Pairwise testing is an approach to combinatorial testing that executes a pairwise test data set.***
- ***Pairwise test data set - A set of test cases that covers all combinations of the selected test data values for every pair of a system's input variables.***

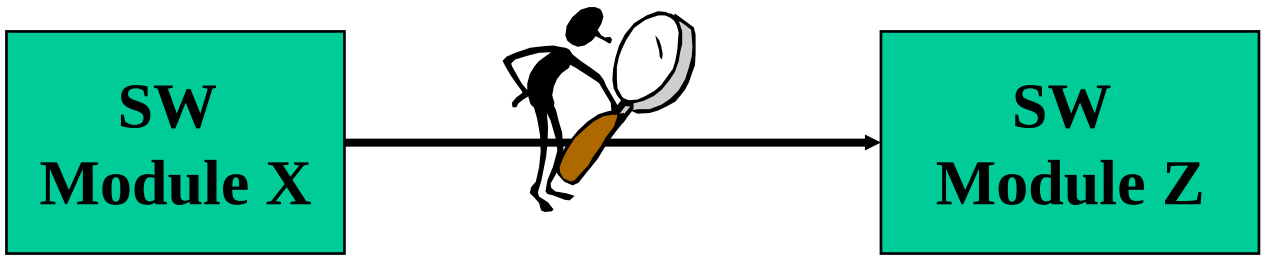
TestNo	DEPENDENTS	MARITAL STATUS	SPOUSE FIRST NAME	EXPECTED RESULT
Test1	No children	Single	Mary	
Test2	No children	Married	Null	
Test3	No children	Divorced	Greater than 20 characters	
Test4	Dependent children	Single	Null	Bug is detected in Test 2
Test5	Dependent children	Married	Greater than 20 characters	
Test6	Dependent children	Divorced	Mary	
Test7	Adult children	Single	Greater than 20 characters	
Test8	Adult children	Married	Mary	
Test9	Adult children	Divorced	Null	

only 9 test cases instead of 27!

ENG.7 - SW Integration Test (3)

Focusing on SW Integration Test is made difficult by:

- ***"What is SW Integration Testing?"***
- ***Poor documentation at SW Design level***
- ***Heavy usage of global variables***



ENG.7 - SW Integration Test (2)

Effective approach may include combination of:

- ***Smoke Testing (based on functional testing)***
- ***Perfom SW Integration Testing to verify that the CCR (Critical Computer Resources) usage is whithin defined thresholds….***

Table 13 — Methods for software integration testing

Methods		ASIL			
		A	B	C	D
1a	Requirements-based test[a]	++	++	++	++
1b	Interface test	++	++	++	++
1c	Fault injection test[b]	+	+	++	++
1d	Resource usage test[c, d]	+	+	+	++
1e	Back-to-back comparison test between model and code, if applicable[e]	+	+	++	++

ENG.8 - SW Testing

SW Testing is made difficult by:

- ***Poor SW Requirement Specification***
- ***Missing Traceability Data***
- ***Not enough Time or Resources***

ENG.8 - SW Testing (2)

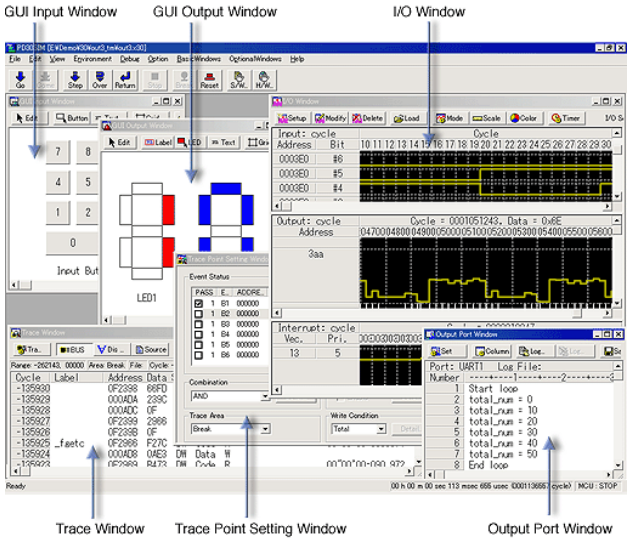
A viable approch for SW Testing is:

- ***First of all Improve SW Requirements Specification Quality***
- ***Then, pursue a Requirement-based Testing approach***
  - ***Trade-off in the specification quality of SW test cases is acceptable***
  - ***Some Automation may be possible depending on the environment - IDEs often allow to collect source code coverage measures***
  - ***Grey-Box Testing: Using an emulator it is typically possible to monitor specific points in SW (e.g. SW interfaces, global variables, registers)***

ENG.8 - SW Testing (3)

For some applications SW Testing can also be executed (and even automatized) conveniently on the Simulator offered by IDE

- ***For example it can be possible to setup a GUI to stimulate the software***



The drawback is that the environment is less representative!

ENG.9 - System Integration Test

Focusing on System Integration Test is difficult because:

- ***"What is System Integration Testing?"***
- ***Poor System Architectural Design***

ENG.9 - System Integration Test

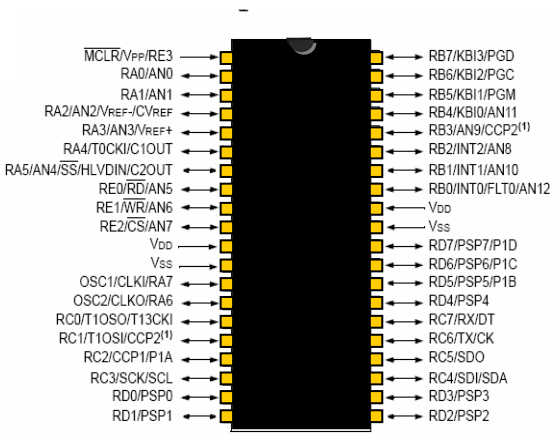
Effective approach may include:

- ***Focus on Interface Testing***
  - ***Focus on HW-SW Interface (HSI) Testing***

  - ***Rely on System Testing and prototypes assembling to complement HSI Testing (HW-MEC, MEC-MEC)***

ENG.9 - System Integration Test (2)

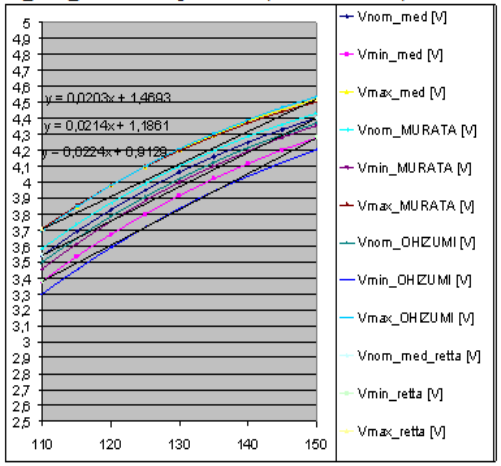
Effective HW-SW Interface Testing needs a HSI detailed specification:



Port A – Digital I/O and Analogue in …
Port B – Digital I/O and Analogue in …
Port C – Serial (RS-232 / SPI/ I2C) …
Port D – Digital I/O …
Port E – More Analogue ….

5.2.5.1.5 uC_SNS_TMP1

Temperature voltage signal generated by the resistor divider including the temperature sensor NTC2. uC_SNS_TMP1 is a signal that depends on the temperature with the following law:



The nominal formula y=0.0214x+1.1861 shall be implemented in the SW.

The HW/SW uC_SNS_TMP1 interface is:

1. Analog Signal
2. Operating range: 1 to 3.0V
3. Worst case range: V_{SS}-0.3V to V_{DD}+0.3V
4. Offset ground level: 0mV (theoretical) at 25°C
5. Offset drift from 0µV/°C
6. Voltage Conversion de-rating: T.b.verified
7. Signal frequency range: equivalent to a time constant of tbd ms

ENG.10 - System Testing

System Test is made difficult by:

- ***Designing and engineering effective system test cases (sometimes can be highly complex)***
- ***Poor System Requirements Specification***
- ***Time Consuming***

Standard system test activity is usually performed by a dedicated group (e.g. LABORATORIO) to test the system in its operating environment

ENG.10 - System Testing (2)

Effective approach to System Testing may include combination of:

- ***Coordination with System Engineers***

  Emphasis on Requirement Verification Criteria

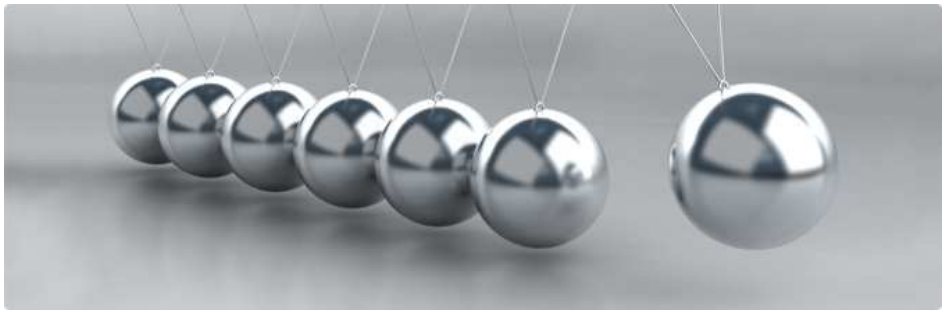
- ***Synergy with corporate quality management (often driven by APQP - Advanced Product Quality Planning)***

  Balance between Functional and Environmental Testing

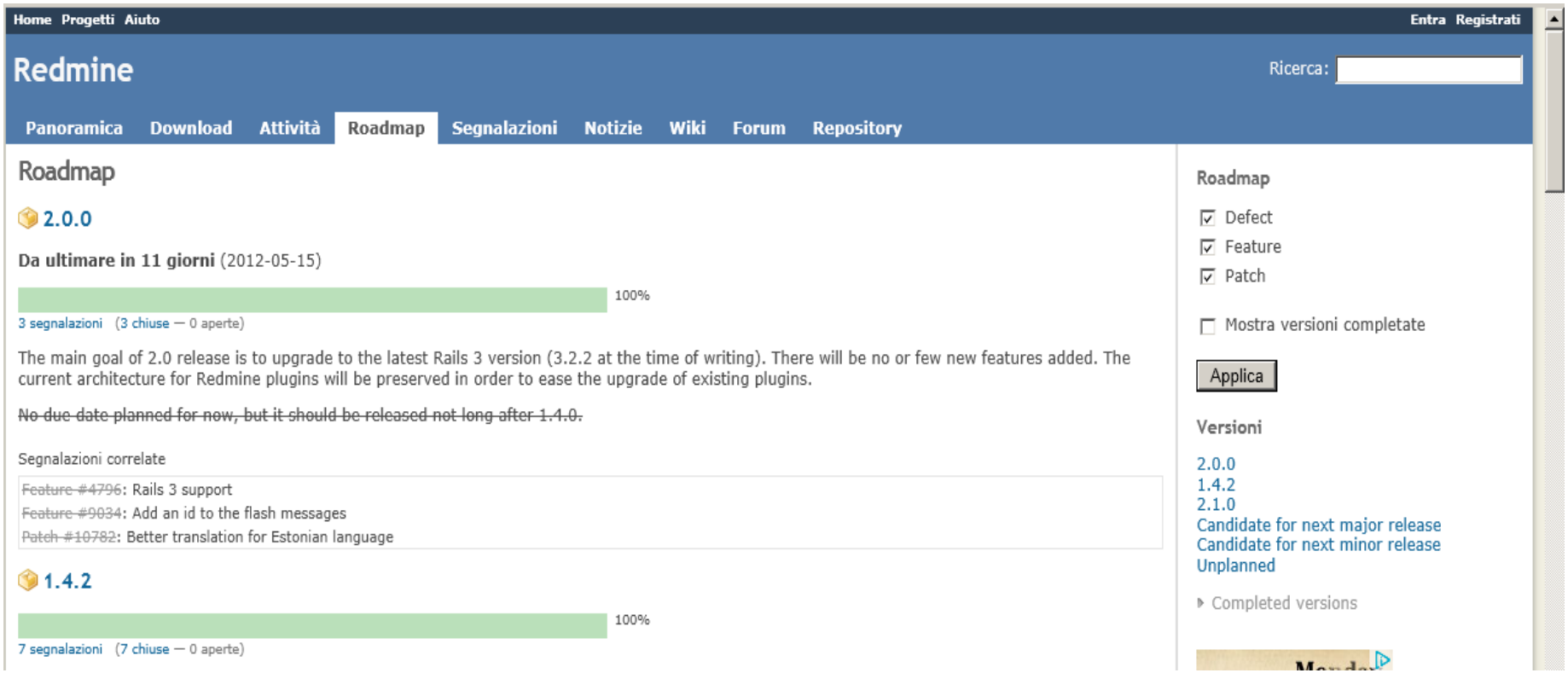
What about regression test strategies?

Effective and Sustainable Regression Testing Strategies may include:

- ***Selective and Flexible selection riteria***
- ***Rely on Good Traceability Data***



...and the planning of V&V tasks?



www.redmine.org

Conclusions

The presented strategies have been implemented (partially or fully) in several "small settings" scenarios, very often with satisfactorily results.

THANKS.

fabio.falcini@intecs.it