

ResilTech

Technologies for Resilience

Safety Analysis at SW level in ISO26262: experience on how to apply FMEA on SW architectures

Automotive SPIN Italia
10° Automotive Software Workshop
25/10/2012
Milan, Italy

Outline

- **Short company introduction**
- **SW FMEA: introduction and motivation**
- **Proposed methodology**
- **Feedback from application**
- **Conclusion and next steps**

Techniques and Technologies for Resilience

- **Company**

- SRL born in late 2007
- Founded by
 - university researchers expert in resilient computing and
 - specialists in the industrial field of Verification and Validation (V&V) of critical systems

- **Mission**

«To provide **engineering consulting and design services** to companies and public bodies mainly for, but not limited to, the field of **resilient systems and infrastructures**»

- **Research**

- Strong relations with both universities and research institutes
- Activities on FP7 projects
- Artemis



ISTITUTO DI SCIENZA E TECNOLOGIE
DELL'INFORMAZIONE "A. FAEDO"

ISTI-CNR (Pisa-Italy)



Università degli Studi di
Firenze (Florence-Italy)

- **Automotive Working groups**

- ISO SC3/WG16 for ISO26262 ("Road vehicles - Functional safety")
- AUTOSAR Phase III
 - WP 1.3 – Safety



- **Automotive Certification**

- Partnership with TUV-SGS for functional safety certifications



SW FMEA: introduction and motivation

- The ISO26262 lifecycle foresees the safety analysis to be performed at different levels: system, SW and HW.
- Identifying and understanding the impact of faults is more and better understood at system and HW levels
 - Here a more “classical” FME(D)A approach can be adopted
 - Note that also in relation to HW metric evaluation a typical inductive, bottom-up procedure can be recognized
- On the contrary a similar activity at SW level has not such an established background.
- Nevertheless this is a required activity
 - and with specific aims,
 - **specific but not very clear!**
- Let's see first to
 - understand why we need this activity and what are the relationships with the overall lifecycle
 - and, consequently, a way to perform the activity itself

SW FMEA: WG16 roadmap

- **WG16 related activities so far**

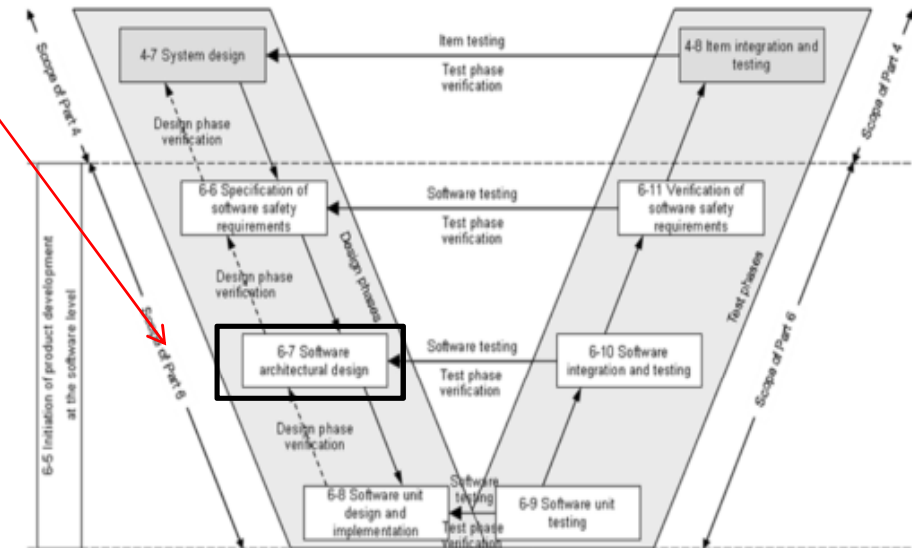
- Issues of SW Safety Analysis issues discussed in Italian group in April 2012
- Concept proposal on SW FMEA elaborated by Resiltech and discussed within Italian delegation
- Italian proposal presented and discussed on Paris meeting on 18/06/2012
 - **High level of interest and good acceptance specially from France delegation**
- Informal work group established and initial concept started to be elaborate

- **and next steps**

- Once a shared and elaborated proposal is present it will be officially made available on ISO repository and discussion will be expanded to all WG16 members.
- So far possible ways of incorporating new material on first revision of ISO26262
 - **Informative examples on methodology on Part 6 and/or Part 10.**

Where and why in the lifecycle

- With reference to the SW lifecycle we are at the SW architectural design stage.
- Within the lifecycle principal aim is
 - to support the specification of safety mechanism at software architecture level.
- Then the output of the activity is
 - to modify the architecture to accommodate error detection mechanisms (and proper reactions of the SW in line with original safety concept).
- and/or
 - provide evidence that existing architecture is completely or partially fine as it is.
 - *Link with system level safety analysis is then somehow automatic/compulsory!*



Activity outcome

- **Some potential outputs could be**

- Range checks of input and output data
- Plausibility check
- Control flow monitoring
- Diverse software design (parallel paths)
- Change data flow/dependency
- *External monitoring facility*
- *Correcting codes for data*

***Extracted from the standard but
more HW related. Take care!***

- **When starting the analysis it is probable that the architecture already accommodate some of the these solutions.**
- **Reason is that they probably derive from SW safety requirements (derived from system level).**
- **In this case outcome can be seen as a providing evidence that these are “good enough” on the real SW architecture.**

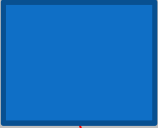
How to do it – Level of details

- **Since the aim is to study the architecture the proper level of details is the architectural level.**
- **This means that SW component in the architecture could (should) be treated as a black box**
- **and the effect of their internal failure reported outside to understand the impact on the architecture and finally on the safety requirements.**
- **This is also in line for model based development, where study has to be done at higher levels.**

How to do it – Modelling SW misbehaviour

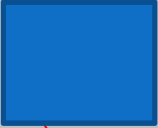
- Analysis could be based on keyword approach to identify and then model the misbehavior of a SW component.
 - E.g. providing a given “Output incorrect”
- Severity of the failure is understood in terms of impact on safety goals (or anyway top level safety requirements allocated to SW).
- Which “faults” are causing the SW to misbehave?
 - *ISO26262*
 - *NOTE Safety mechanisms can be specified to cover both issues associated with random hardware failures as well as software faults.*
 - *Method is to take into account only SW faults for this analysis.*

How to do it – Analysis flow

Failure identification		System level mapping	Failure impact on Safety Goal	Safety Mechanisms spec	Conclusion
					

- Correlated each SW component with the failure mode keywords.
- Include special (fake) components to include impact of calibration data and corrupted input signals.
 - "Calibration data" component
 - this is needed to trace impact of the corruption of data
 - "Input signals" component
 - this is needed to check the impact on the architecture of signals coming from external world

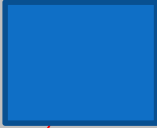
How to do it – Analysis flow

Failure identification		System level mapping	Failure impact on Safety Goal	Safety Mechanisms spec	Conclusion
					

- **Examples of the failure mode keywords**

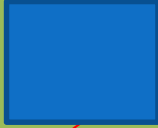
- *Output incorrect*
- *Fail to execute*
- *Output timing incorrect (e.g. execution too long)*
- *Blocking shared resources*
- *Blocking execution of other processes*
- *Access not authorized resources*
- *Corrupted in range input*
- *Corrupted out of range input*

How to do it – Analysis flow

Failure identification		System level mapping	Failure impact on Safety Goal	Safety Mechanisms spec	Conclusion
Apply failure modes to SW component					

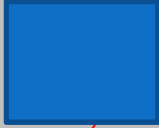
- Description of Specific Failure mode instance

How to do it – Analysis flow

Failure identification		System level mapping	Failure impact on Safety Goal	Safety Mechanisms spec	Conclusion
Apply failure modes to SW component	Description of Specific Failure mode instance				

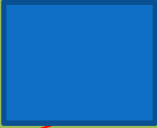
- If a similar failure instance already considered ad system level check impact an actual SW architecture; mainly verify if system level conclusions still hold.

How to do it – Analysis flow

Failure identification		System level mapping	Failure impact on Safety Goal	Safety Mechanisms spec	Conclusion
Apply failure modes to SW component	Description of Specific Failure mode instance	If a similar failure instance already considered at system level check impact an actual SW architecture; mainly verify if system level conclusions still hold.			

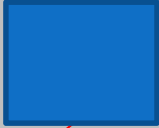
- Severity is judged on impact on Safety Goals or derived high level SW Safety Requirements.

How to do it – Analysis flow

Failure identification		System level mapping	Failure impact on Safety Goal	Safety Mechanisms spec	Conclusion
Apply failure modes to SW component	Description of Specific Failure mode instance	If a similar failure instance already considered ad system level check impact an actual SW architecture to see if system level conclusions still hold.	Severity is judged on impact on Safety Goals or derived high level SW Safety Requirements.		

- Show if already existing SW are efficient as on-line mitigations or specify new ones.

How to do it – Analysis flow

Failure identification		System level mapping	Failure impact on Safety Goal	Safety Mechanisms spec	Conclusion
Apply failure modes to SW component	Description of Specific Failure mode instance	If a similar failure instance already considered at system level check impact an actual SW architecture to see if system level conclusions still hold.	Severity is judged on impact on Safety Goals or derived high level SW Safety Requirements.	Show if already existing SW are efficient as on-line mitigations or specify new ones.	

- Specify if a new SW requirement is expected or not.

Feedback from application ^{1/2}



- **Positive**

- With respect to input signals and system level FMEA
 - Verify system level FMEA
 - Improve confidence/provide evidence of completeness of system level study
 - Verify efficiency of SW technical requirements for Safety Mechanisms
 - Verify/improve efficiency of safety mechanisms at SW level
 - **Example: task of the watchdog refresh**
 - Highlight increase severity of some inputs in relation to how the SW architecture is using the variable for instance in ECU state management
- Highlight potential conflict of shared resources usage/interference
 - Example: timeout strategy
- Highlight potential issues on timing constraints
- and
 - **one “uncovered” failure requiring a new mechanisms (plausibility check) to be implemented!**

Feedback from application ^{2/2}



- **Negative**

- Some kind of SW architecture formalization is needed (this should not be negative...)
 - Different interpretation of the SW architecture
 - Results may "vary" depending on the level of details of the SW description
- Once a potential issue is found it is not always straightforward to motivate usage of some error detection and mitigation techniques versus a SW testing strategy.
 - E.g. 1 - Intermediate output variables corrupted: why not "simply" testing enough?
 - E.g. 2 - Adopt SW diversity: when does it make sense?

Conclusion and next steps

- It is important to make it clear (or rather agree on) what is meant for a SW safety analysis.
- Basic of the proposed approach is quite well accepted so far and it seems a good starting point.
- **Next steps**
 - Come up with a shared list of keywords for failure modes and suggest possible interpretation.
 - Consider maybe specific and different lists for application and basic SWs. So far the target is primarily the application SW.
 - Share a case study and “work by example”.



Thanks for your attention!

francesco.rossi@resiltech.com