PHILIPS

sense and simplicity

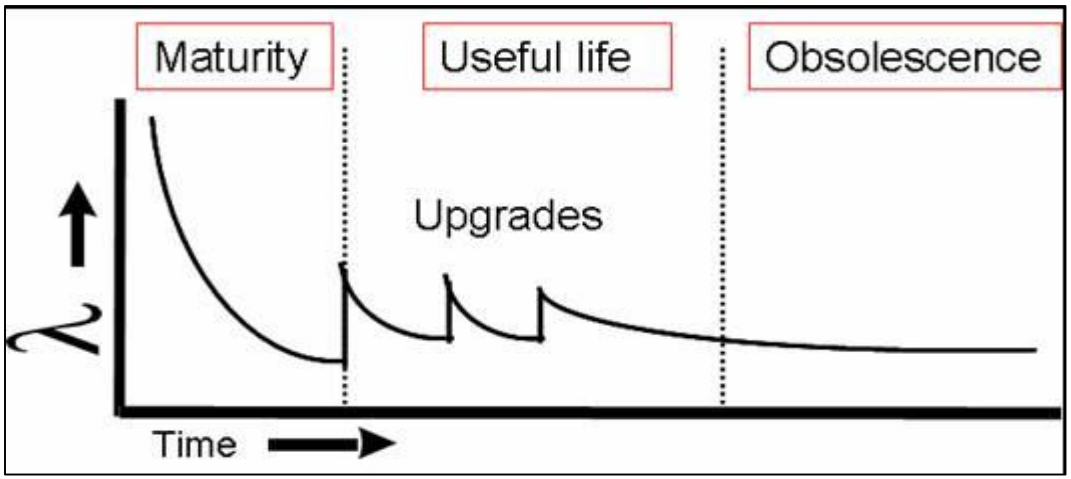
Reliability : Software Engineering Perspective - ISSRE 2009

Ajit Ashok Shenvi
Philips - Bangalore
Sep 23, 2009

PHILIPS

Background

- Software Reliability is an important factor in *System* Reliability as the contribution of software in products is constantly increasing
- *Software Reliability is the probability of failure-free software operation in a specified environment for a specified period of time (or natural units)*
- NO ONE uniform theory of software reliability yet. NO ONE widely accepted method of estimating or predicting software reliability yet
- Software by itself does not have a “Constant Failure rate (random failures)”, hence defining MTBF for only software is tricky
- The typical bath-tub curve for software would look something like this :



PHILIPS

Software Reliability – Contents of this paper

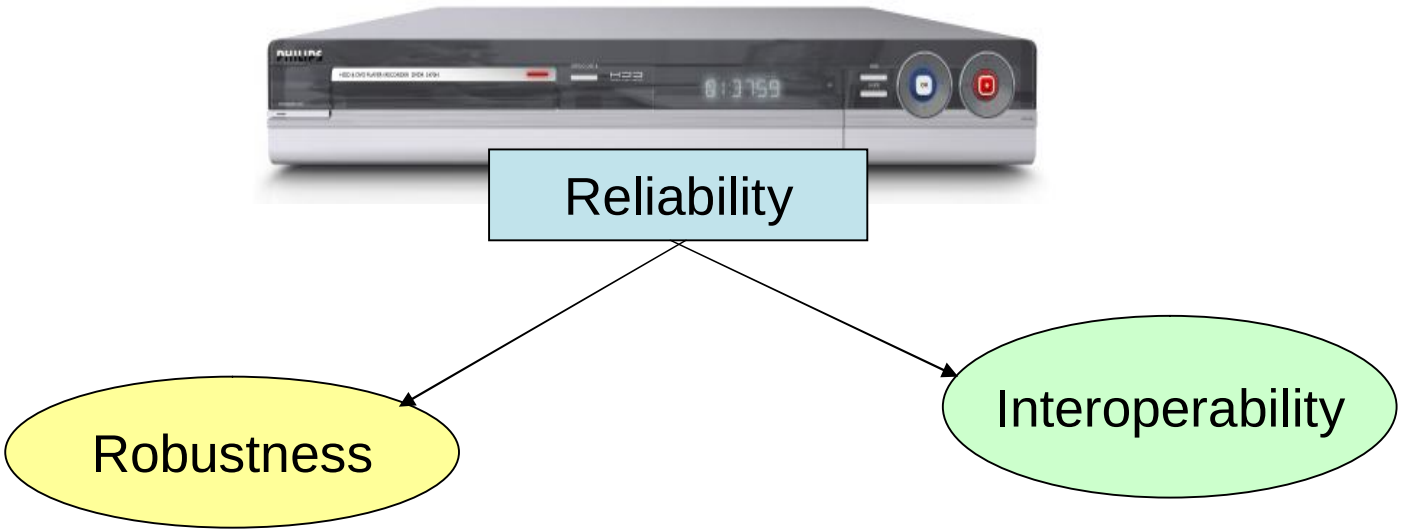
- Theme - *Building in* Software Reliability during the development life cycle

- Case study of DVD-Hard disk recorder product –
  - definition of Reliability as *"Critical to Quality (CTQ)"* in terms of Robustness and Interoperability aspects
  - flow down of the CTQs into the lower parameters
  - design-in to achieve the desired reliability from software

- Recommendation of a framework for ensuring and tracking software reliability along the software development life-cycle
  - built around the three dimensions of *Fault-Prevention, Fault-Tolerance and Fault-Detection*
  - process structure of CMMI & Orthogonal defect classification, augmented with FMEA and principles of graceful exit mechanisms, supplemented with "Pondering maturity index" for reliability growth.

PHILIPS

Software Reliability – DVD-Hard disk recorder



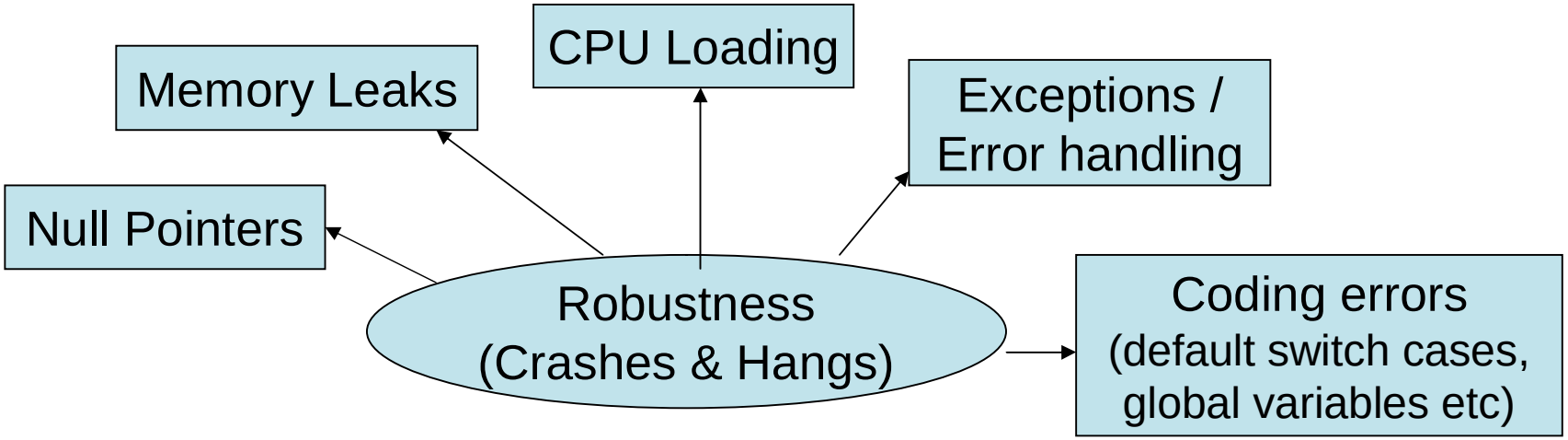
- For this product, Reliability was defined on the following 2 axes:
  - *Robustness* (How often does it hang or crash in normal user scenarios)
  - *Interoperability* (Does it work seamlessly with other devices especially Digital cameras via USB port).

PHILIPS

Software Reliability – CTQ-1 : Robustness

- The CTQ of Robustness was quantified as ***"Number of Hangs/crashes"*** with target as 0
  - in normal scenarios with typical use cases
  - in certain stressed situations with "concurrent" use cases

The lower level factors or (Xs) that could impact the CTQ (Y) Robustness



Robustness = ***f*** (Null pointers, Mem leaks, CPU load, Exceptions, Coding errors)

Targets were set for each of the above parameters.

PHILIPS

Software Reliability – CTQ-1 : Robustness

- A small Script was developed to find all ***"Null pointers"*** in the code stack. These were then eliminated

- Stringent ***Limits*** set for ***memory allocation*** of subsystems. This was tracked at every release to ensure that all subsystems are within budget and that there is no overlap of memory space. (Subsystems come from different project teams and external suppliers)

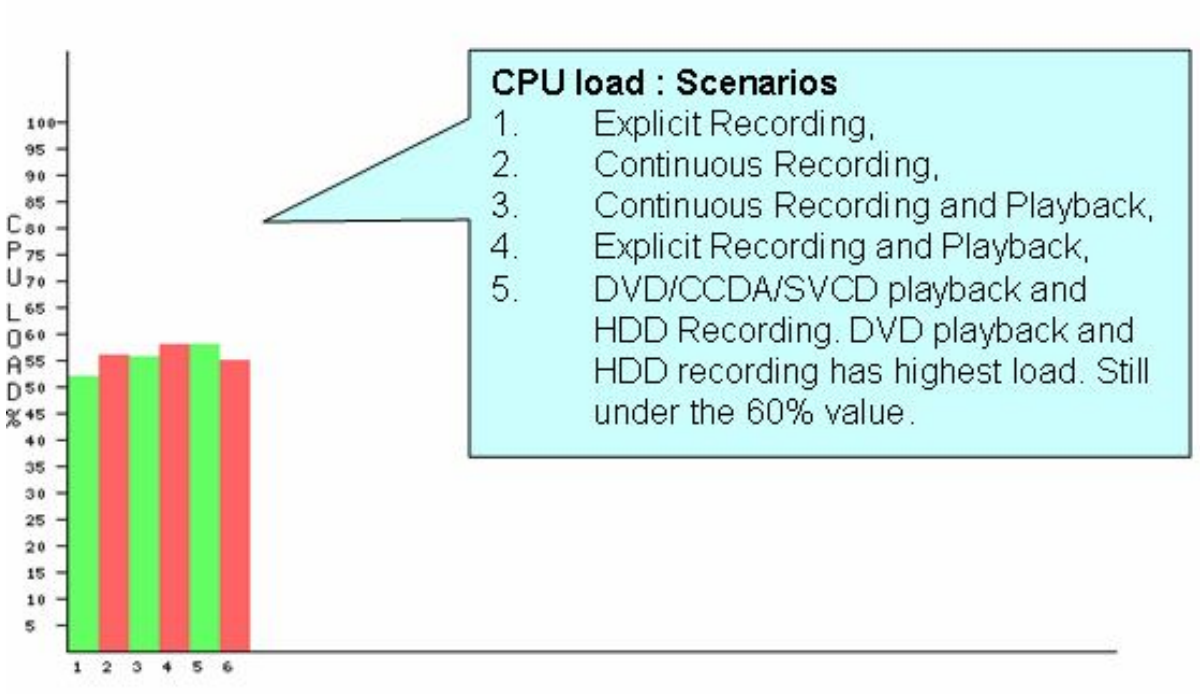
- A script was made to check for implementation of "***default conditions***" for "switch case" statements

- *Static analyzer* tools such **QAC** was run and the target set was 85% code coverage. Errors and warnings were closed.

PHILIPS

Software Reliability – CTQ-1 : Robustness

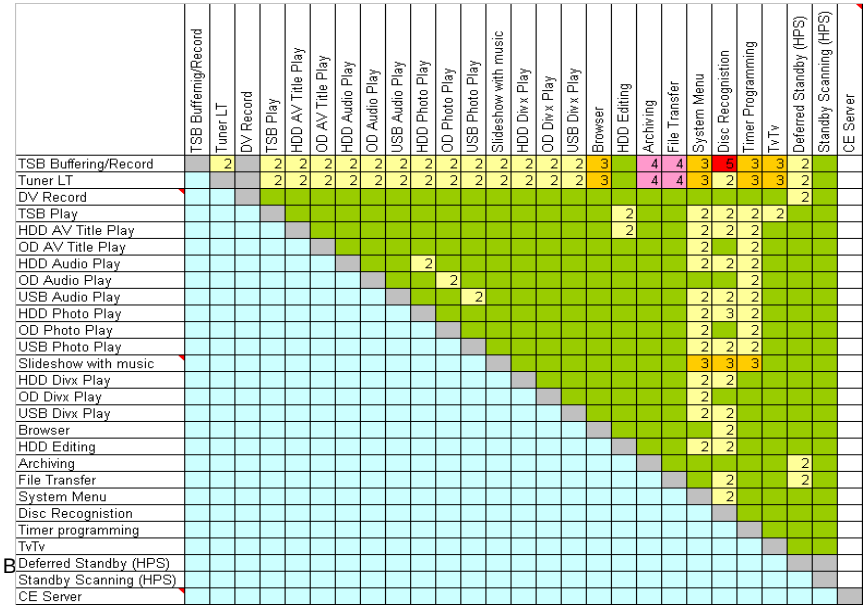
- From Embedded programming experience, it is known that **CPU load > 65%** makes the system unstable and unpredictable
- Different combinations of scenario's (*stressed conditions*) were chosen and CPU load tracked using a tool called CodePerf for every release.



PHILIPS

Software Reliability – CTQ-1 : Robustness

- **FMEA** was done to identify failure modes leading to exceptional conditions for new features.
  - *Graceful exits* and error recovery mechanisms were implemented. For e.g. exit with an error message rather than be in a loop when a non-standard USB device is connected to the recorder, error recovery when a non standard format disc is played on the device
  - A *"concurrency"* matrix (shown below) was made that depicted levels of parallel use cases that could be executed by the user. The initial requirements were simplified so that the crash and hang conditions could be reduced.

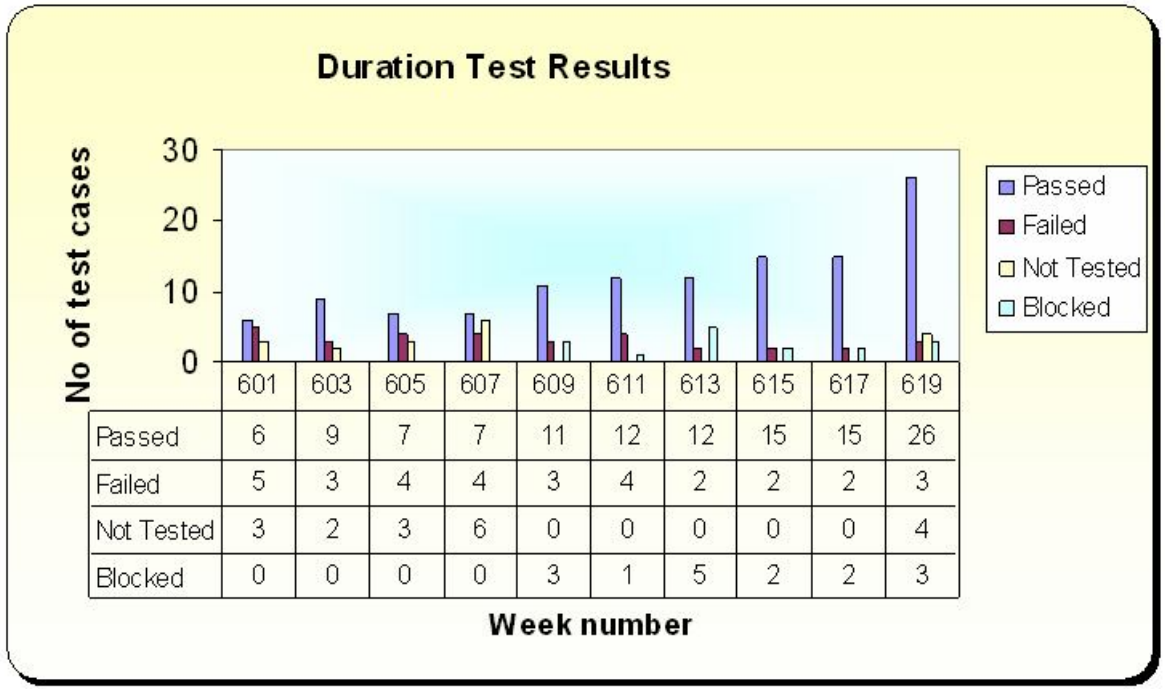


PHILIPS

Software Reliability - CTQ-1 : Robustness

- An *operational profile* of typical user scenarios created and the software run on product with different profiles in a continuous loop for 4-days at elevated temperatures (Duration test)

- The results verified every alternate weeks on every build.

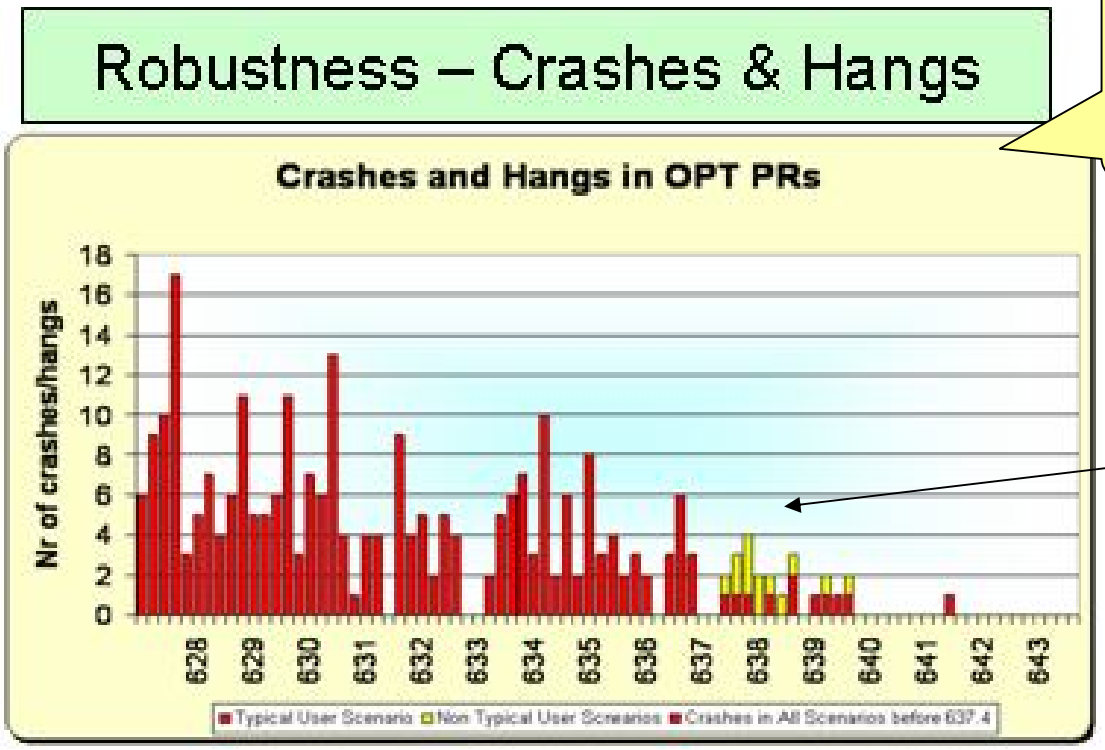


	601	603	605	607	609	611	613	615	617	619
Passed	6	9	7	7	11	12	12	15	15	26
Failed	5	3	4	4	3	4	2	2	2	3
Not Tested	3	2	3	6	0	0	0	0	0	4
Blocked	0	0	0	0	3	1	5	2	2	3

PHILIPS

Software Reliability - CTQ-1 : Robustness

- Finally the CTQ of robustness – hangs and crashes were measured on weekly basis to verify the results.

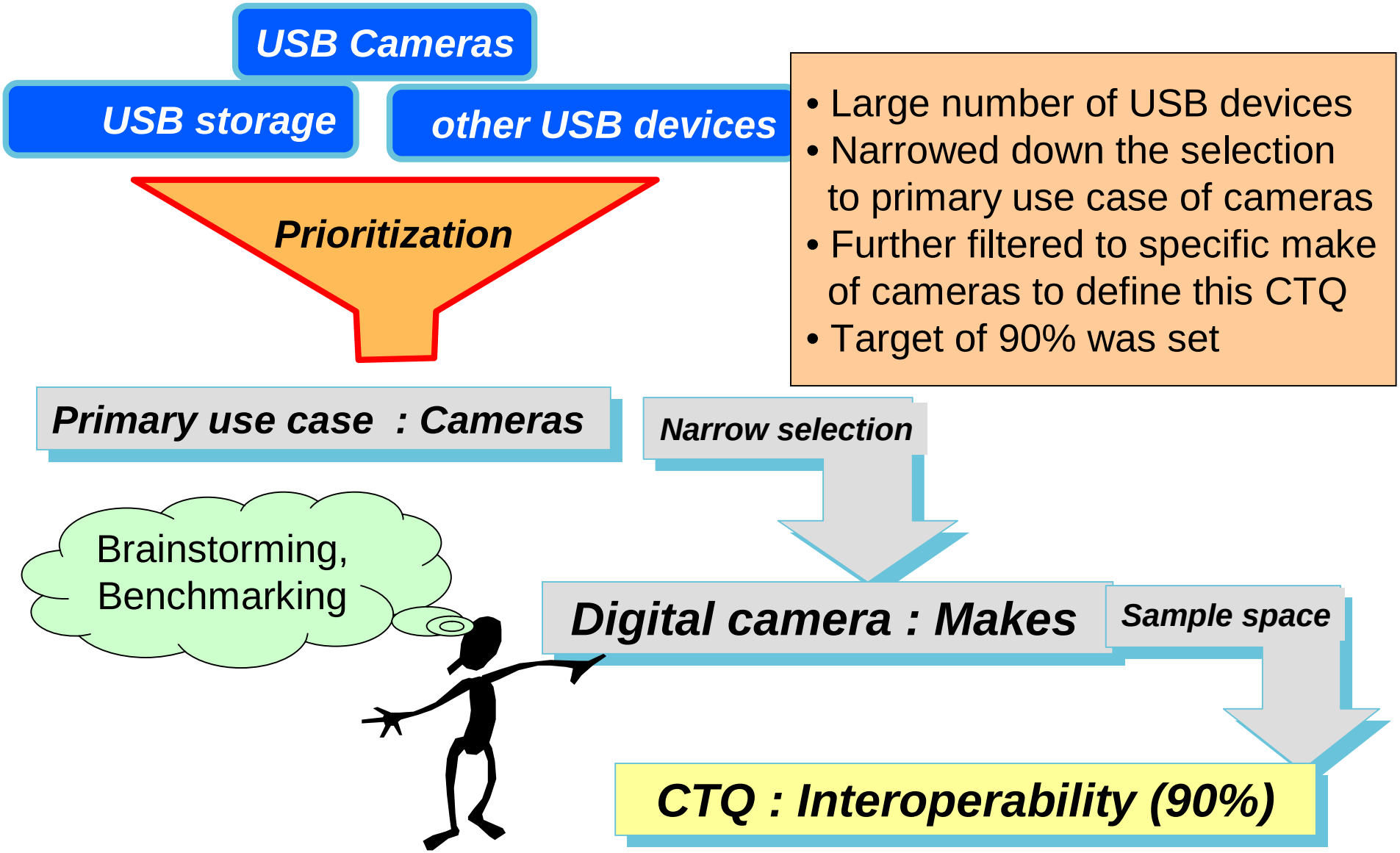


FMEA, Null pointer removal, Default condition checks, QAC, Error recovery, memory tracking, Duration tests from pre-integration, CPU load tracking, border cases etc

Assert related crashes

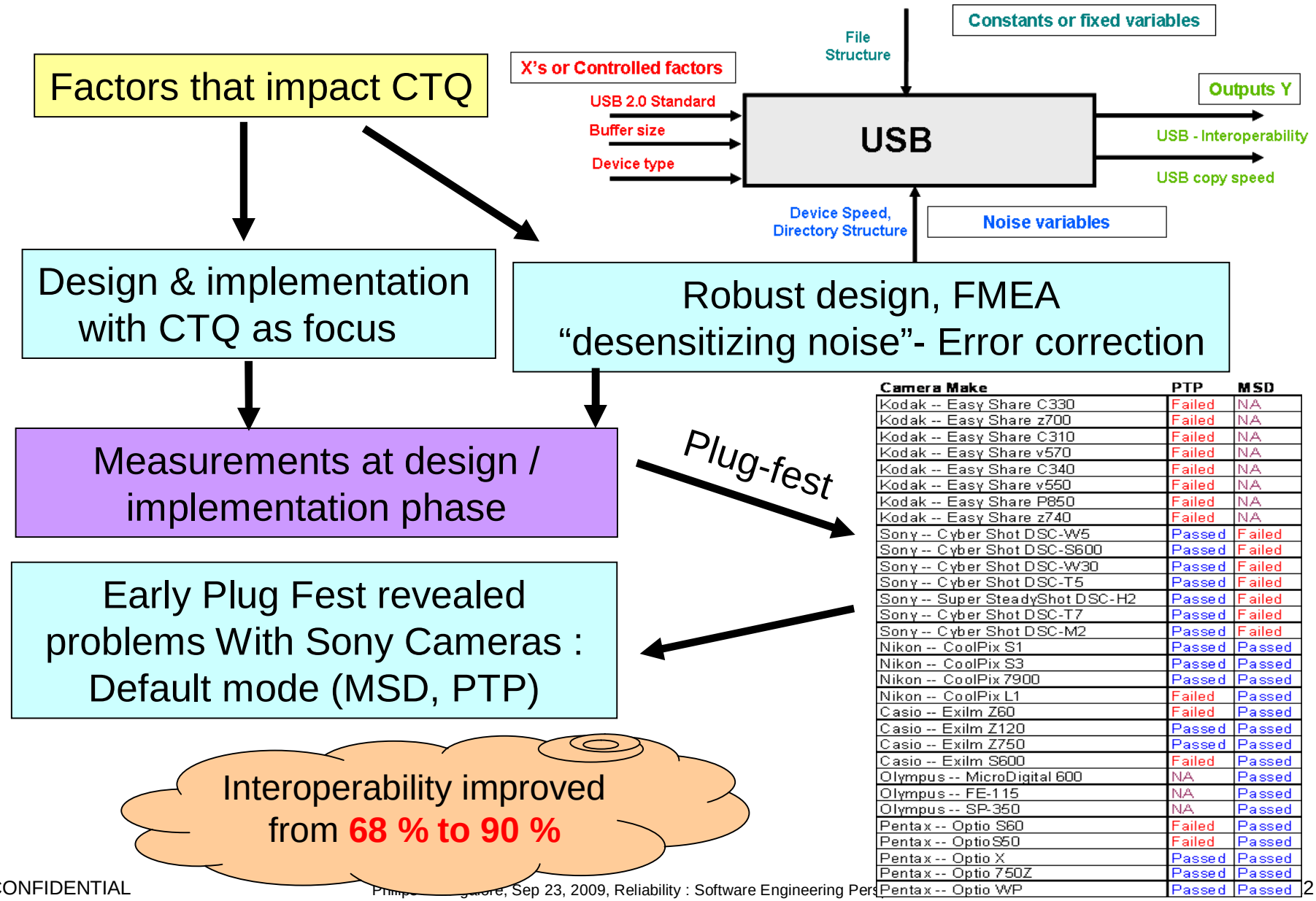
PHILIPS

Software Reliability - CTQ-2 : Interoperability



PHILIPS

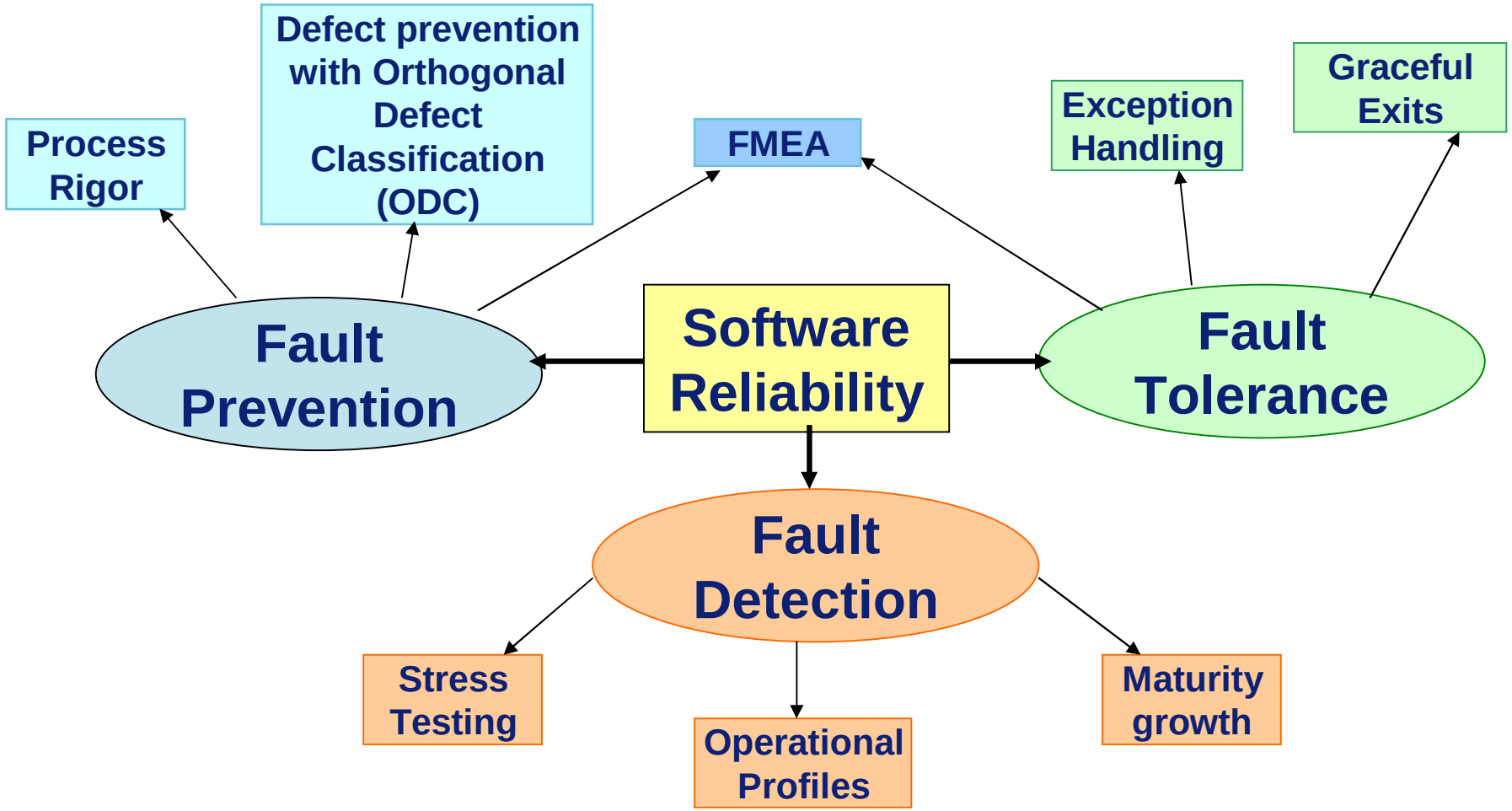
Software Reliability – CTQ-2 : Interoperability



Camera Make	PTP	MSD
Kodak -- Easy Share C330	Failed	NA
Kodak -- Easy Share z700	Failed	NA
Kodak -- Easy Share C310	Failed	NA
Kodak -- Easy Share v570	Failed	NA
Kodak -- Easy Share C340	Failed	NA
Kodak -- Easy Share v550	Failed	NA
Kodak -- Easy Share P850	Failed	NA
Kodak -- Easy Share z740	Failed	NA
Sony -- Cyber Shot DSC-W5	Passed	Failed
Sony -- Cyber Shot DSC-S600	Passed	Failed
Sony -- Cyber Shot DSC-W30	Passed	Failed
Sony -- Cyber Shot DSC-T5	Passed	Failed
Sony -- Super SteadyShot DSC-H2	Passed	Failed
Sony -- Cyber Shot DSC-T7	Passed	Failed
Sony -- Cyber Shot DSC-M2	Passed	Failed
Nikon -- CoolPix S1	Passed	Passed
Nikon -- CoolPix S3	Passed	Passed
Nikon -- CoolPix 7900	Passed	Passed
Nikon -- CoolPix L1	Failed	Passed
Casio -- Exilm Z60	Failed	Passed
Casio -- Exilm Z120	Passed	Passed
Casio -- Exilm Z750	Passed	Passed
Casio -- Exilm S600	Failed	Passed
Olympus -- MicroDigital 600	NA	Passed
Olympus -- FE-115	NA	Passed
Olympus -- SP-350	NA	Passed
Pentax -- Optio S60	Failed	Passed
Pentax -- OptioS50	Failed	Passed
Pentax -- Optio X	Passed	Passed
Pentax -- Optio 750Z	Passed	Passed
Pentax -- Optio WP	Passed	Passed

PHILIPS

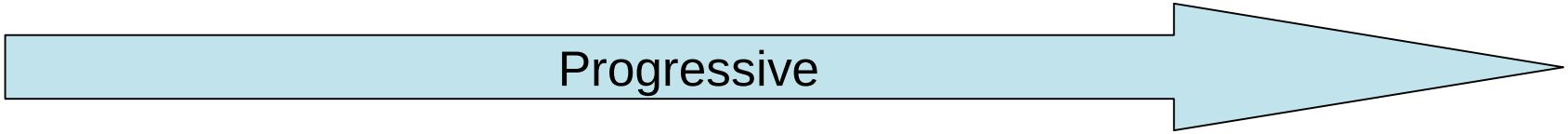
Software Reliability Framework - Elements



- Software Reliability framework can be built around the 3 pillars - Fault Prevention, Fault Tolerance and Fault Detection.

PHILIPS

Means to ensure Reliability

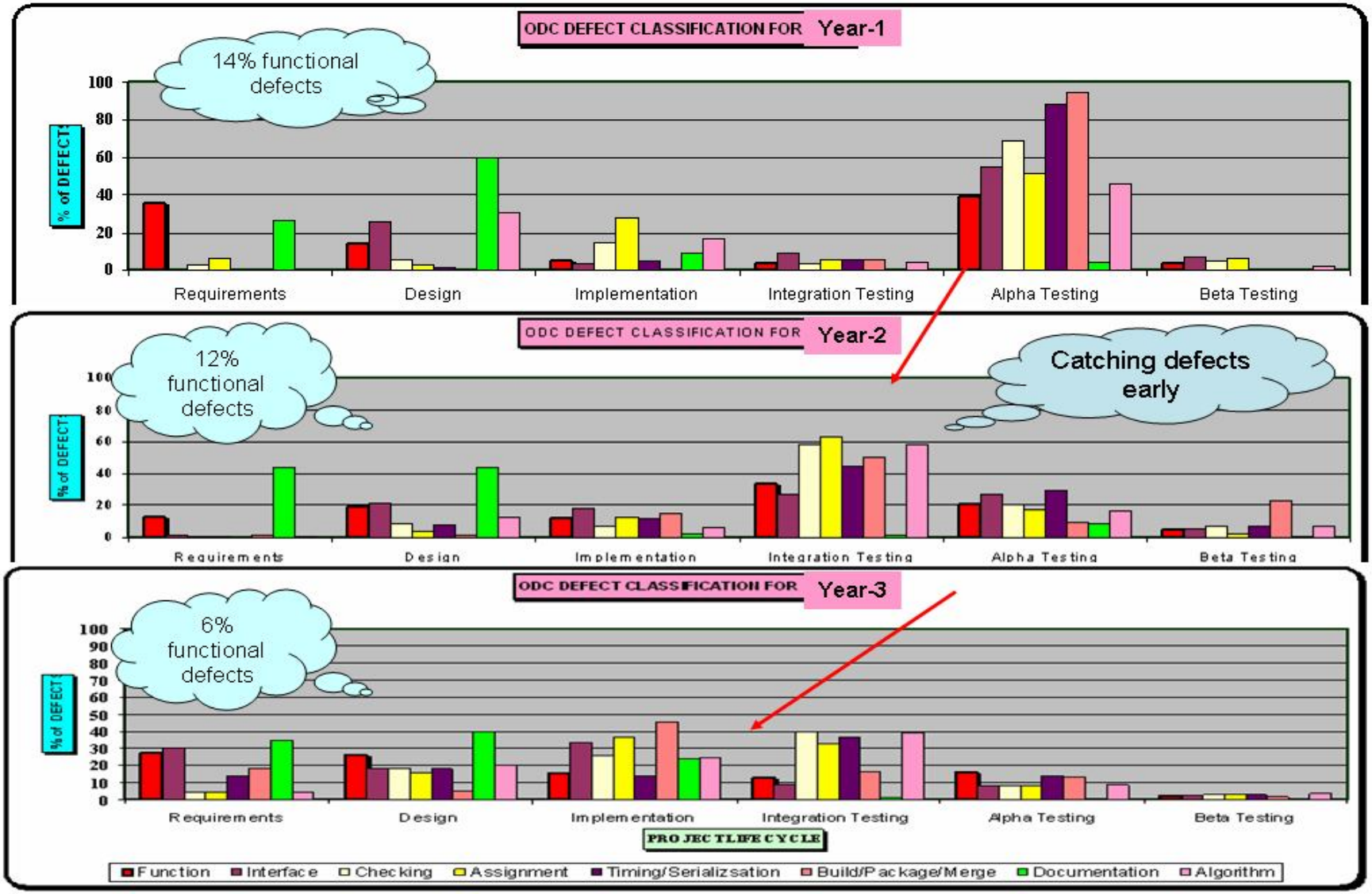


Fault Prevention	Fault Tolerance	Fault Detection
• Process Rigor (CMMI) – Requirements Management with CTQs defined around Voice of Customer – Architecture, design with CTQs as the basis – Strict enforcement of Coding guidelines – Bidirectional Tracebility – Peer and expert Reviews • ODC based Defect prevention • FMEA and Mistake proofing	• FMEA • Exception Handling • Graceful exits • Graceful degradation	• Operational profiles testing • Stress tests and accelerated tests • Reliability growth with Pondering maturity Index

PHILIPS

Means to ensure Reliability – Fault Prevention

Defect Prevention Structure built around ODC (Orthogonal Defect Classification).



PHILIPS

Means to ensure Reliability – Prevention /Tolerance

- FMEA is an excellent tool to build-in fault tolerance and fault prevention mechanisms in systems based on CTQs and use cases.

Recommendation	If your FMEA will have more than 30 components, process steps or product functions, perform FMEA simplification first. See Tab titled "Simplification"												Action Results				
Sl No	Module/ Component/ Functionality / Feature	Potential Failure Mode	Failure Effect	SEV	Potential Causes	OCC	Current Controls	DET	RPN	Actions Recommended	Resp.	Target Date	Actions Taken	SEV	OCC	DET	RPN
Unique identification	Module name, Function, Features, Process Step	What can go wrong?	What is the impact of the failure?	How Severe is the effect to the customer?	What are the causes?	How often does cause or FM occur?	What are the existing controls and procedures (inspection and test) that prevent either the cause or the Failure Mode?	How well can you detect cause or FM?		What are the actions for reducing the occurrance of the Cause, or improving detection?	Who is Responsible for the recommended action?	What is the target date	What are the completed actions that were taken?				
	TSB																
	TSB Marking	Begin marking is done by the user. End marking is not done .	User loses the whole recording or some part of the recording	6	User did a operation without closing the marking	5	Requirements/Design	2	60	Record till end of program if program information is available or record till the next virtual title or record till 6hrs	Mani	706	Requirements have been changed to specify these new rules	6	1	3	18
		Marking/Recording lost	User loses the whole recording or some part of the recording	6	No space on HDD		Requirements/Design			While marking if there is no place on the HDD then	Mani		Requirement	6	1	2	12

The RPN – Risk priority number is a good indicator of reliability – Higher the RPN, lower the reliability

PHILIPS

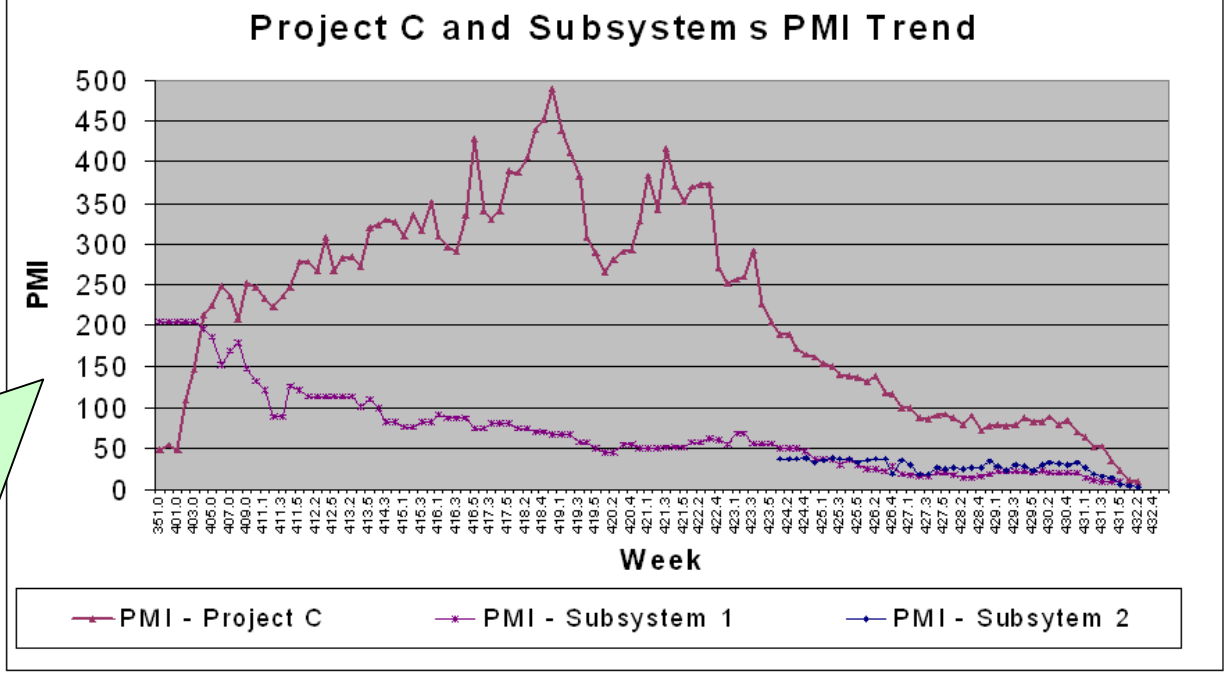
Means to ensure Reliability – Fault Detection

		Maturity Grid						
State of PR	Severity/ Evolution	S	A	B	C	D	PMI	Release Decision
Submitted	4	0	5	4	2	0	106.125	'NO'
In_analaysis	3	0	11	30	7	1		
analysed, in_resolution	2	0	38	130	30	5		
resolved, in_evaluation	1	0	81	63	27	8		
Other	0							

Note:
1. PMI = MI/40 where,
MI= 80*(S4) + 60*(S3) + 40*(S2) + 20*(S1)
+ 40*(A4) + 30*(A3) + 20*(A2) + 10*(A1)
+ 20*(B4) + 15*(B3) + 10*(B2) + 5*(B1)
2. "Other" includes - 'Evaluated', 'Hold', 'PT rejected', 'Equal to', 'Not reproducible'.
3. Target PMI < 5
4. No events in red area

- "Pondering Maturity Index (PMI)" is a number computed based on Severity of bugs and their evolution
- Weightage is allocated for each combination to calculate overall PMI

This PMI can be used to track Reliability growth during the development phase and a decision point during release



PHILIPS

Software Reliability – Summary

- Reliability in simple terms implies *"failure free operation"*
- Definition of a software failure will vary depending on the kind of product
- It is important to define what "software reliability" means for a particular product based on Voice of the customer
- Using this as CTQ, it can be flowed down into the architecture and design
- Reliability can be ensured by : Fault prevention, Fault tolerance and Fault detection techniques built into the software engineering framework
- Some leading indicators than can help estimate "Reliability"
  - Risk priority number from FMEA
  - PMI value in conjunction with Test coverage
  - Pre-release defect density
  - Mean-time between Crashes and Hangs
  - Process compliance scores from CMMI

PHILIPS

References

- Jiantao Pan. 1999. Software Reliability. Carnegie Mellon http://www.ece.cmu.edu/~koopman/des_s99/sw_reliability
- *Chillarege, R., Bhandari, I. S., Chaar, J. K., Halliday, M. J., Moebus, D. S., Ray, B. K. & Wong, M., Orthogonal Defect Classification: Concept for In-Process Measurement, IEEE. Transaction on Software Engineering, Vol. 18 No. 11, Nov-92 http://www.chillarege.com/odc/articles/odcconcept/odc.html*
- CMMI® for development v1.2. August 2006 Reference : CMU/SEI-2006-R-008 ESC-R-2006-008CMMI Product team http://www.sei.cmu.edu/reports/06tr008.pdf
- Pondering Maturity Index : SPEED, Software Creation Process – Philips Consumer Electronics, UAT-0482-3301


PHILIPS