Projects in Real Life

THE UNIVERSITY OF MELBOURNE

Case Study of Project Risk
Siemens – R & D Metrics

Prepared by Ferenc Birloni, PhD

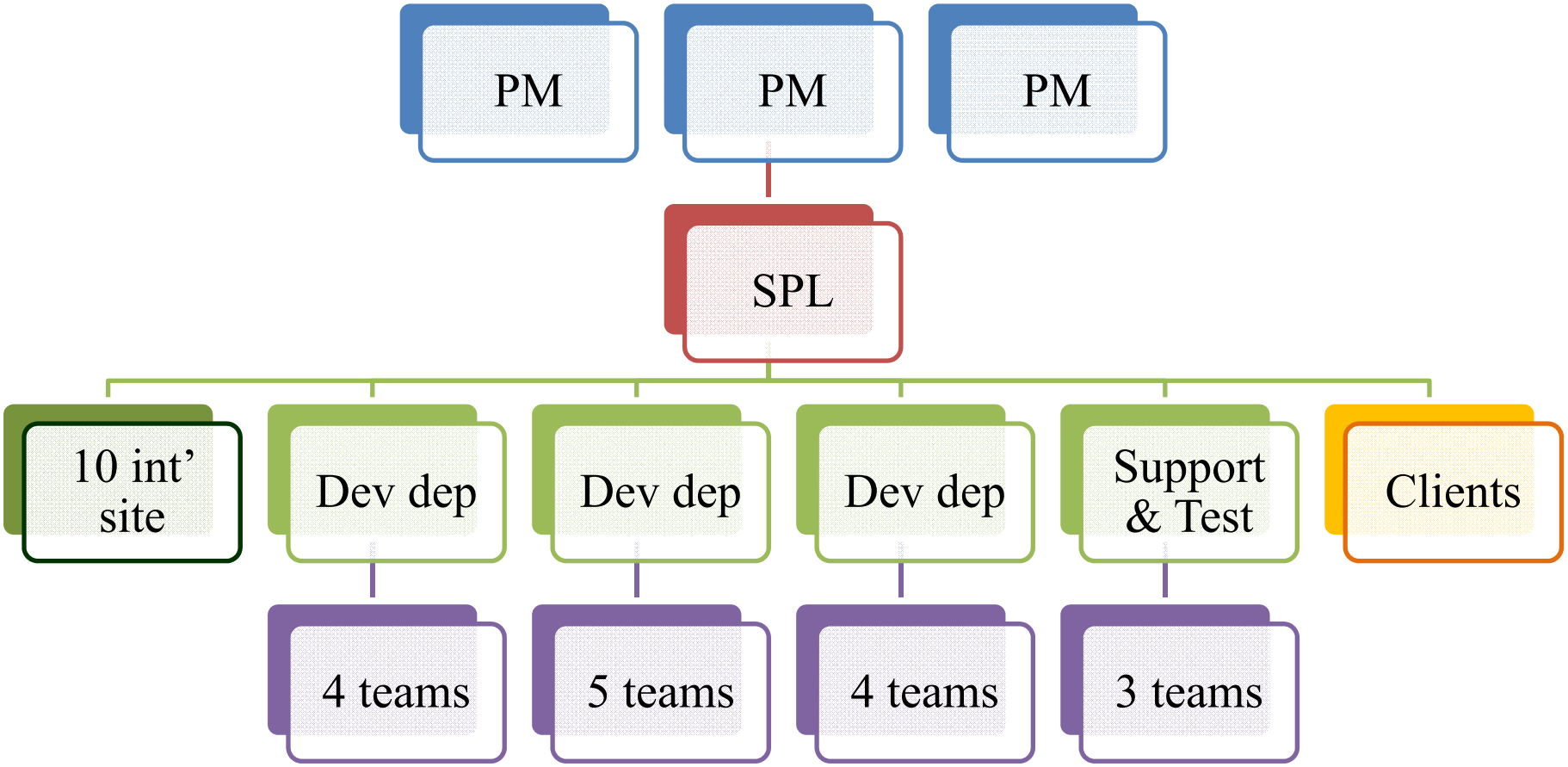
2017

THE UNIVERSITY OF MELBOURNE

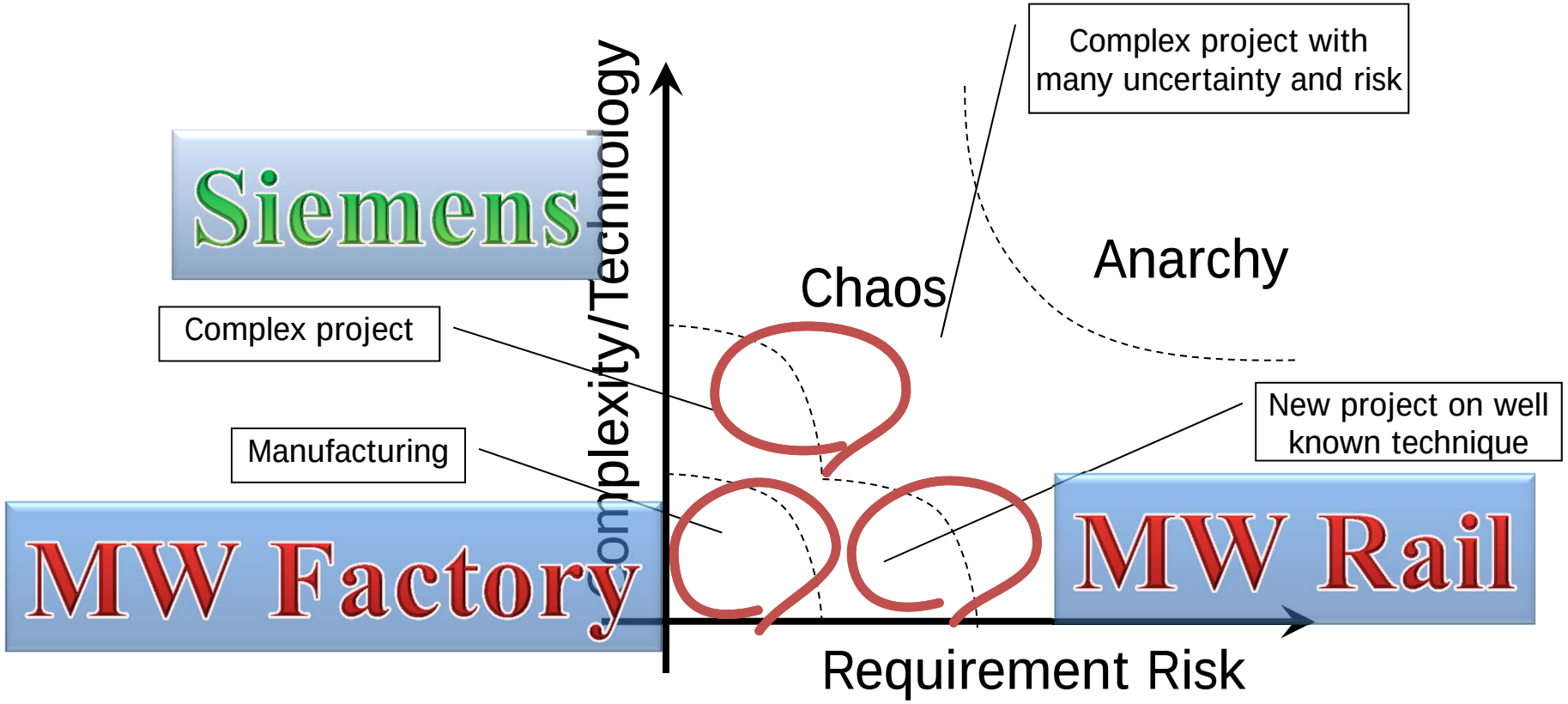
RISK ON DOCUMENTATION

- ✓ Can earn 3 times the contracted value?
- ✓ Electrical, Computer or IT Engineers?

Siemens


PM
PM
PM
SPL
10 int' site
Dev dep
Dev dep
Dev dep
Support & Test
Clients
4 teams
5 teams
4 teams
3 teams

Project Risk Level Assessment



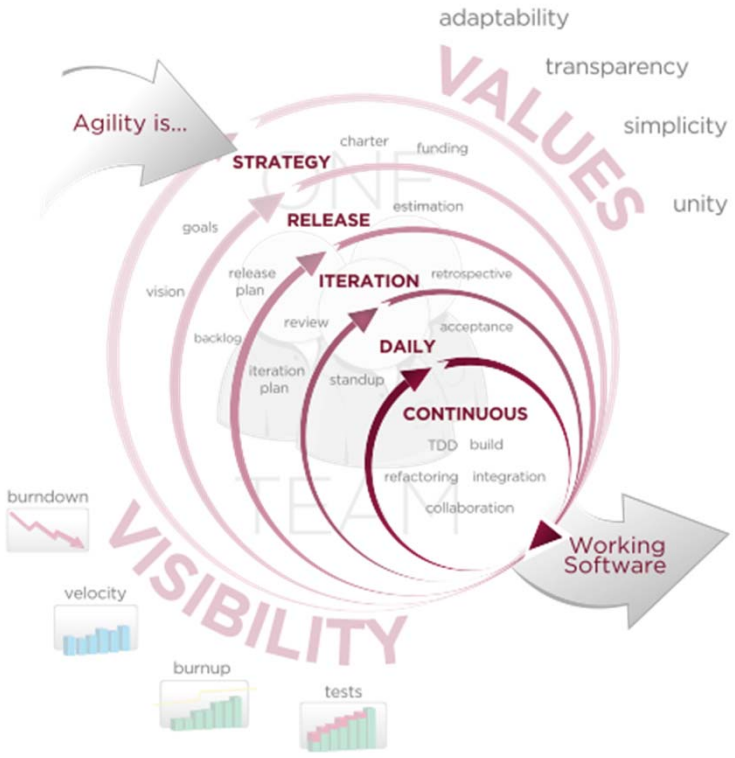
THE UNIVERSITY OF MELBOURNE

SIEMENS – ENTERPRISE

- ✓ Risk – Enterprise Vertical & Horizontal Overview
- ✓ Redesign Crises – Case Study
- ✓ Metrics @ Enterprise Project – Case Study
- ✓ Monitoring, Control & Review – Examples

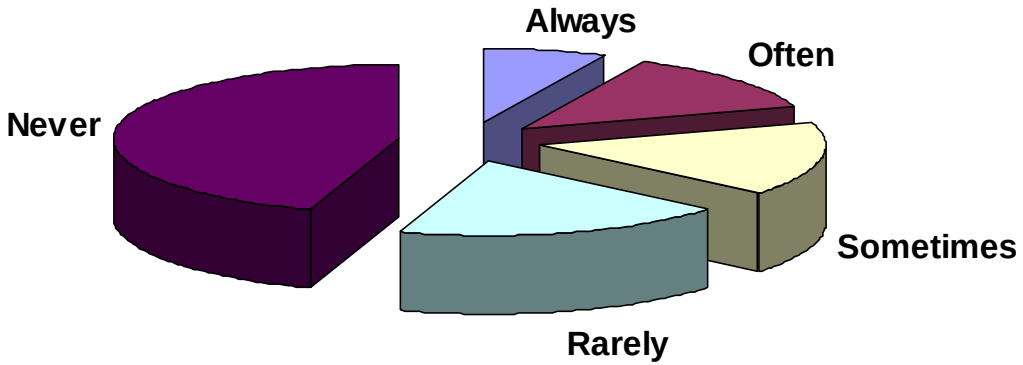
Vertical Risk Levelling

- Risk – by organisational level
- Do not underestimate risk on project execution level
- “Butterfly effect”
- Enterprise – Requires a
  - General but
  - Efficient Risk Metrics

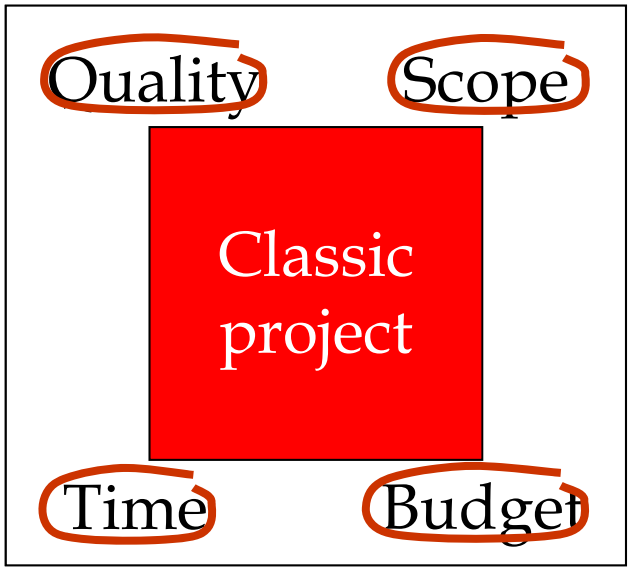


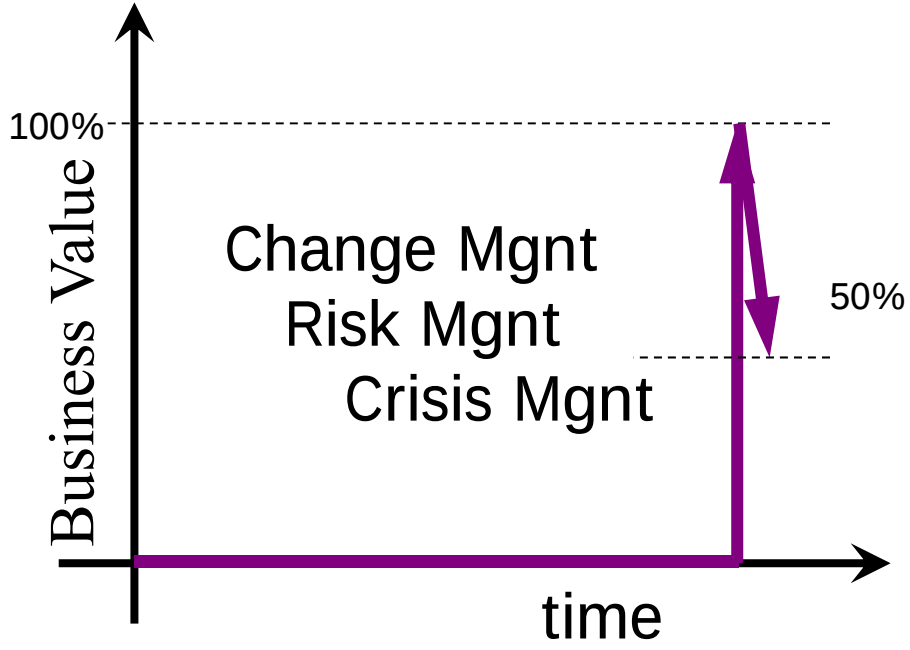
source: http://commons.wikimedia.org/wiki//

Business Value Creation

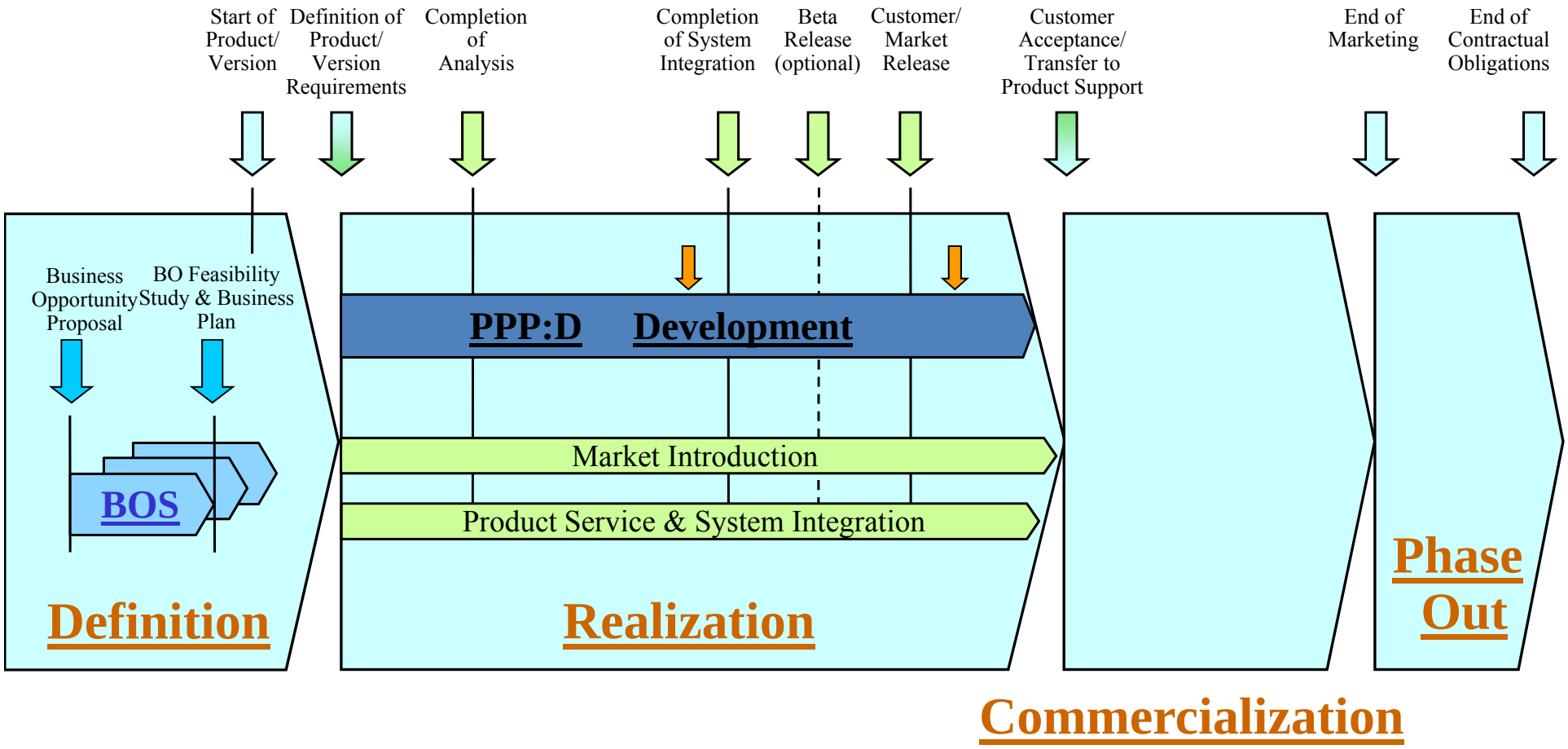


Standish Group Study, 2002

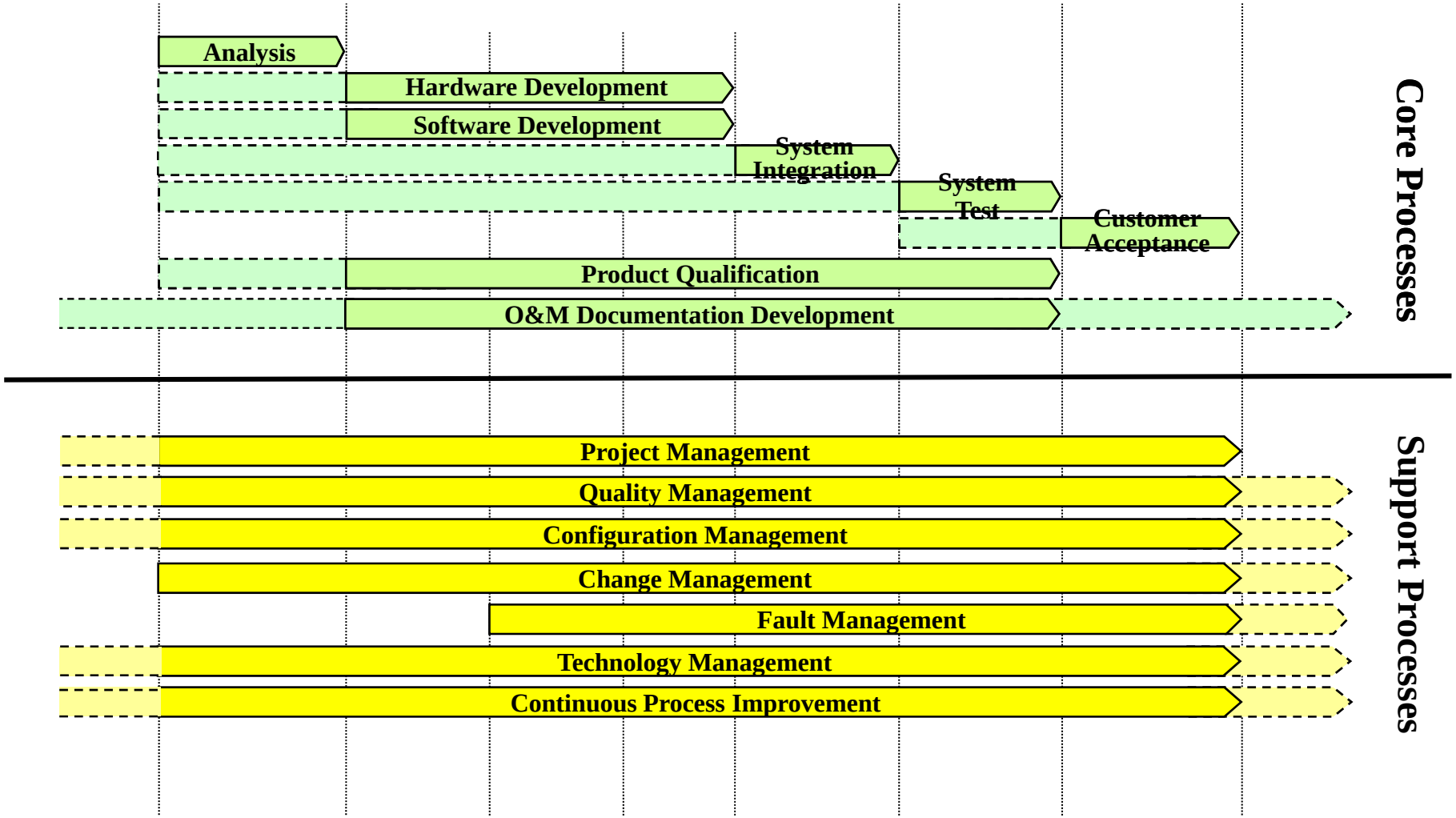




R & D Product Risk



Risks in Process Structure



Herbert

THE UNIVERSITY OF MELBOURNE

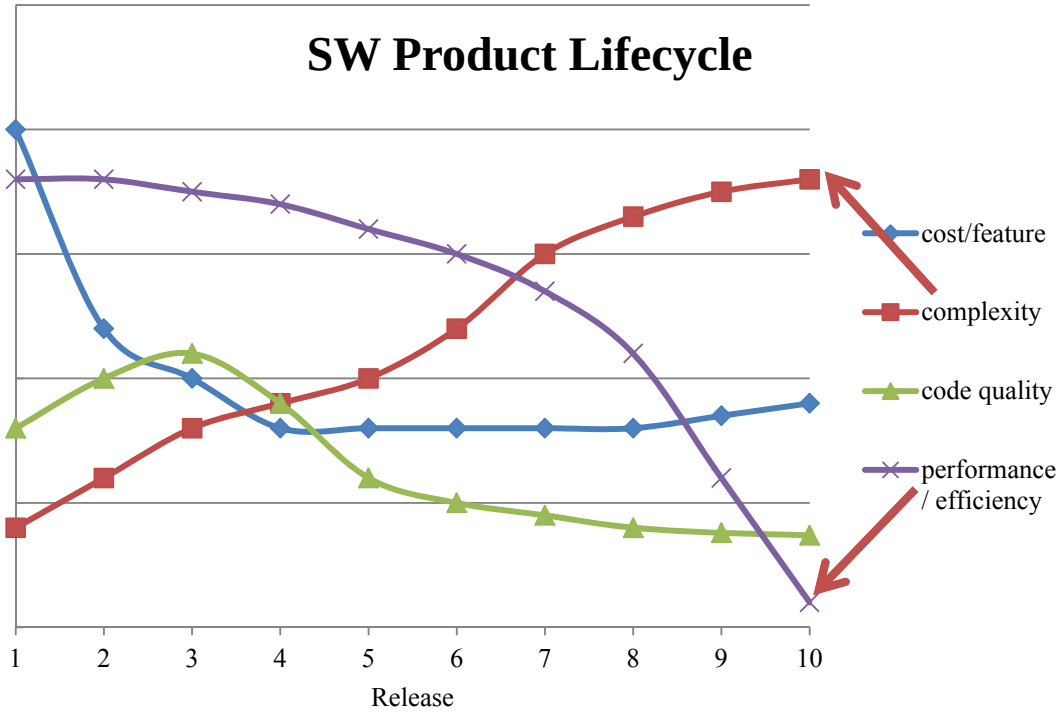
THE UNIVERSITY OF MELBOURNE

REDESIGN CRISES – CASE STUDY

Background – Time to Redesign

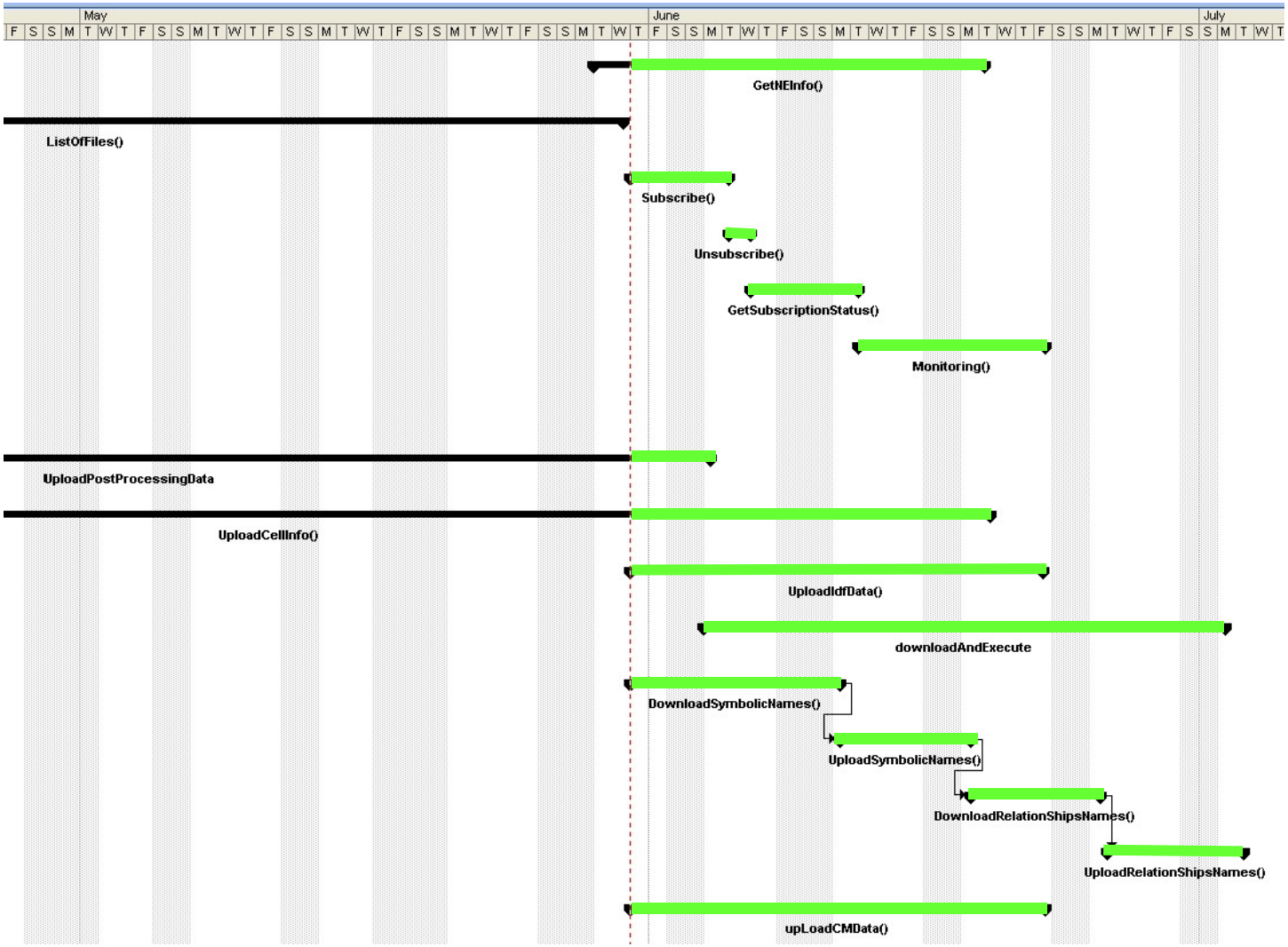
THE UNIVERSITY OF MELBOURNE

- Incremental innovation in development
- New features added on existing code
- Again & again…
- Clean up / garbage
- **Technology** change
- **Architecture**
- Partial rework – more complex
- **Redesign** / start from scratch

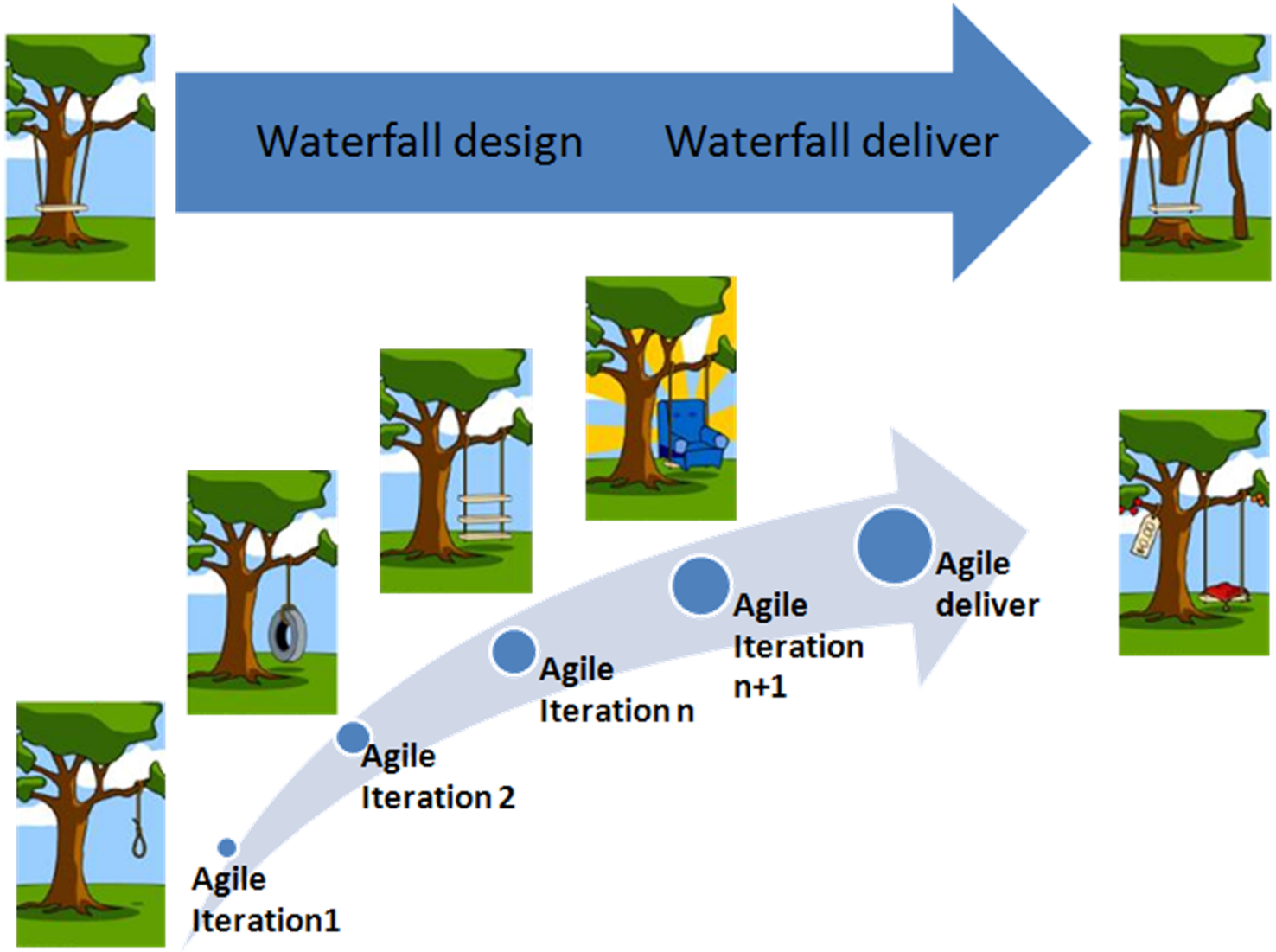


When Redesign Leads to Crisis

THE UNIVERSITY OF MELBOURNE



Active Risk Management – Process



Risk – The Human Factor

THE UNIVERSITY OF MELBOURNE

- How to quantify risk on Human Errors?

source: http://blog.softheme.com/the-future-of-testing-and-the-human-factor/

THE UNIVERSITY OF MELBOURNE

METRICS @ ENTERPRISE PROJECT

- ✓ How much cost to plan for error correction?

Metrics Definition

THE UNIVERSITY OF MELBOURNE

Error Density (ED)

$$\mathbf{ED} = \frac{\mathbf{faults}}{\mathbf{release\ effort}}$$

- ED defines the number of faults (Prio1 – Prio4) per staff month.
- ED is a metric to measure the fault density.

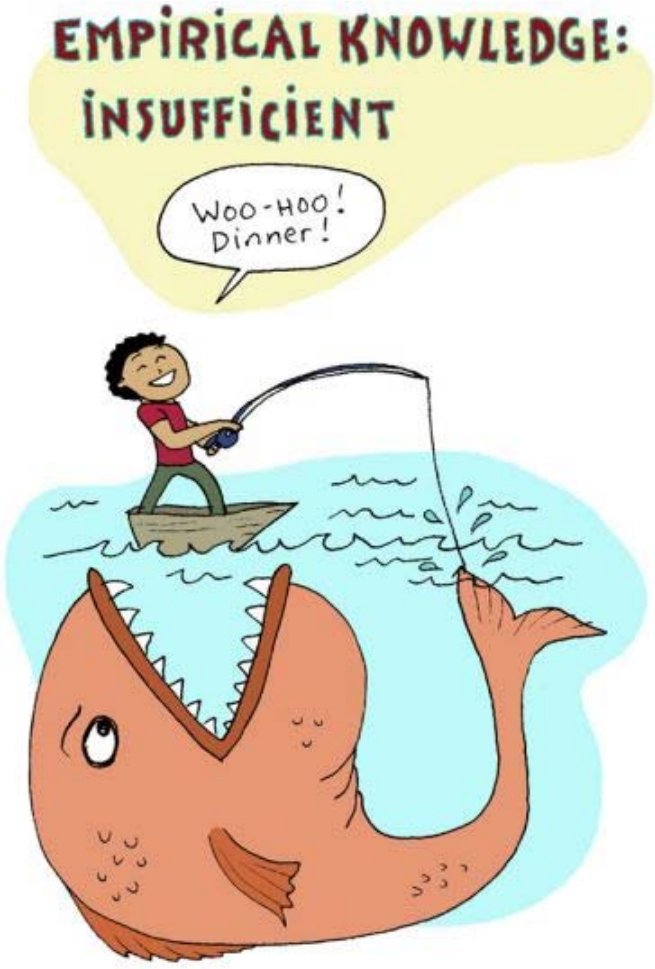
source: http://www.keyconsulting.ca/2013/02/22/fail-to-avoid-failure/

Set ED for Risk Assessment

THE UNIVERSITY OF MELBOURNE

- ED is set at Project Development Kick Off
- **ED=10** (faults/1 man month effort)
- Empirical value

	[MM]
Feature Related Effort	*388*
Change Request buffer	*104*
SUM	**492**
# errors/[MM] effort	10
all # errors	**4920**



source: http://stephaniemcmillan.org/2012/10/

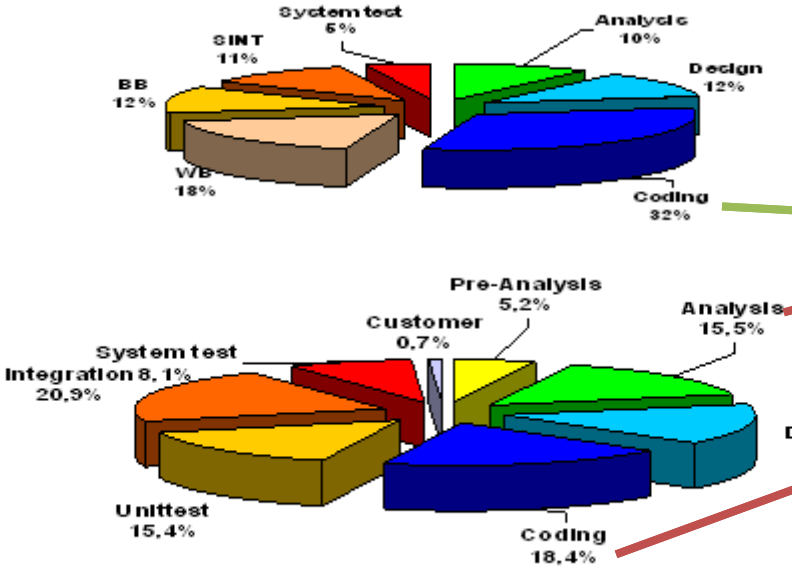
Risk Assessment – 2nd step

Error distribution (experimental)	[%]	corr. time [hours]	expected # of errors Feature #errors	CR #errors	all #errors
Analysis		-			
Design		-			
Coding (comp & func test)		7			
WBT - FTG		7			
BBT		7			
SINT		7			
ST		7			
TACC		10			
SUM	100.0%	-	3880.0	1040.0	4920.0

- Calculated the total # of expected errors
- Need to know how is the error distributed among the phases
- Empirical (and target) distribution will set the expected # of errors for each phase

Empirical Fault Distribution

- Fault Distribution shows the distribution of faults over phases (milestones)
- Target Distribution set, based on past performance



Error distribution (experimental)	[%]
Analysis	*13.0%*
Design	*13.0%*
Coding (comp & func test)	*30.0%*
WBT - FTG	*15.0%*
BBT	*9.0%*
SINT	*8.0%*
ST	*8.0%*
TACC	*4.0%*
SUM	**100.0%**

Risk Assessment – 3rd step

THE UNIVERSITY OF MELBOURNE

Error distribution (experimental)	[%]	corr. time [hours]	expected # of errors			Feature error corr. [MM]	CR error corr. [MM]	All error corr effort [MM]
			Feature #errors	CR #errors	all #errors			
Analysis	*13.0%*	-	504	135	**640**	-	-	**-**
Design	*13.0%*	-	504	135	**640**	-	-	**-**
Coding (comp & func test)	*30.0%*	*7*	1164	312	**1476**	56.68	15.19	**71.87**
WBT - FTG	*15.0%*	*7*	582	156	**738**	28.34	7.60	**35.94**
BBT	*9.0%*	*7*	349	94	**443**	17.00	4.56	**21.56**
SINT	*8.0%*	*7*	310	83	**394**	15.12	4.05	**19.17**
ST	*8.0%*	*7*	310	83	**394**	15.12	4.05	**19.17**
TACC	*4.0%*	*10*	155	42	**197**	10.80	2.89	**13.69**
SUM	**100.0%**	**-**	**3880.0**	**1040.0**	**4920.0**	**143.05**	**38.34**	**181.40**

- Expected # of errors calculated from the total with the distribution target based on the empirical statistics
- Effort to allocate for fault correction is estimated by mean correction time and # of faults
- Effort/budget estimation to update, **Risk budget: 182MM**

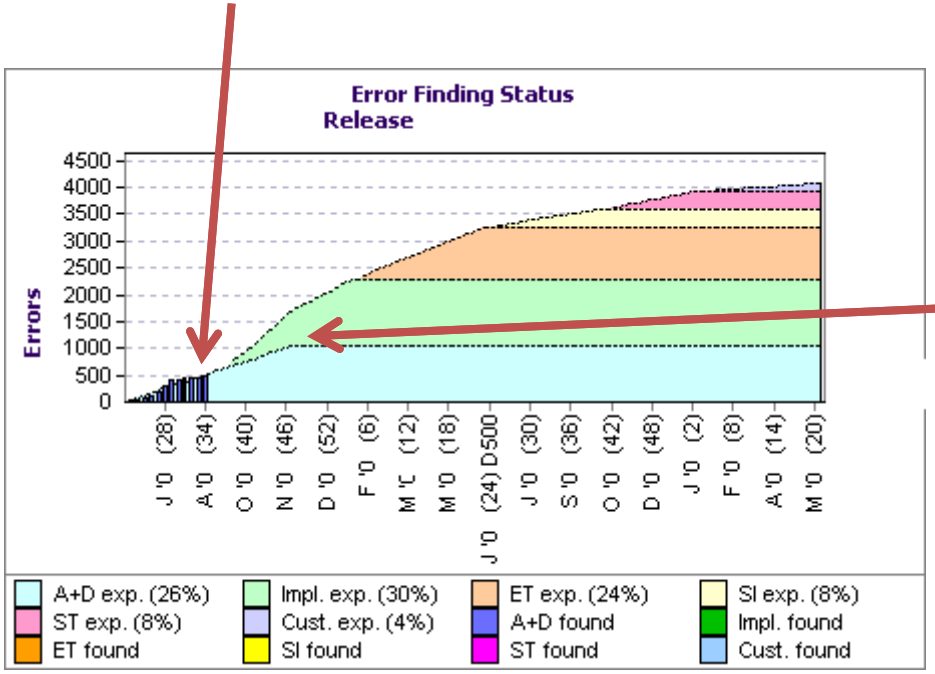
THE UNIVERSITY OF MELBOURNE

MONITORING, CONTROL & REVIEW

Error Finding Status (EFS)

- Progress Tracking

	GSM	
Feature Effort	380	3880
Change Requ	104	1040
Sum:	484	4920

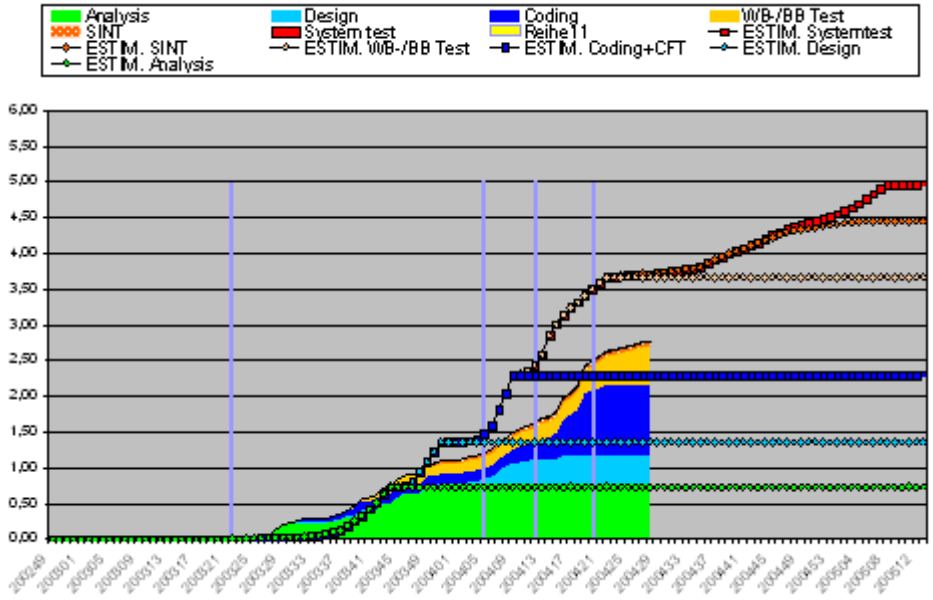


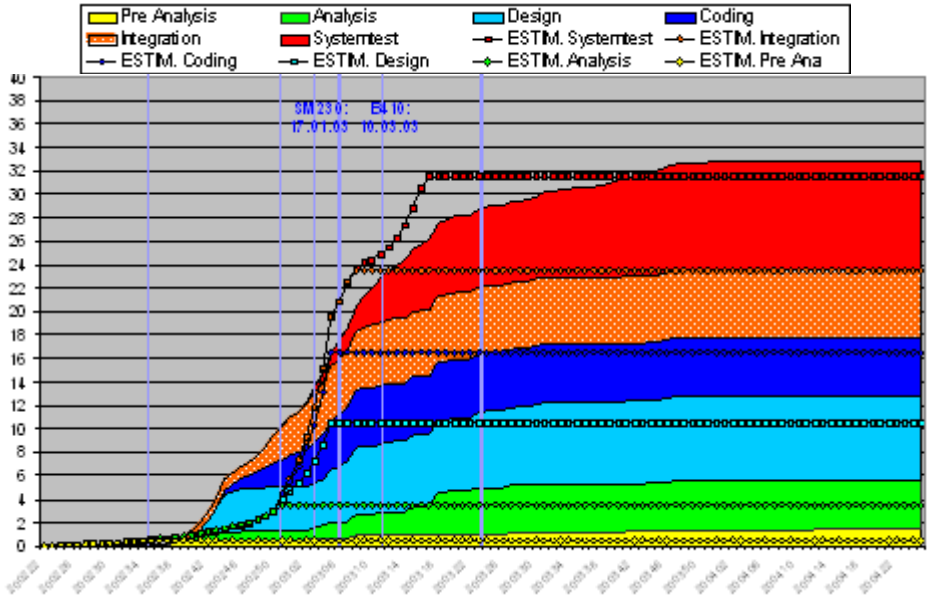
RC 9.0	Target Distribution	Errors to be found
Analysis	13%	640
Design	13%	640
Implementatic	30%	1476
WBT	15%	738
BBT	9%	443
SINT	8%	394
ST	8%	394
Field	4%	197
Sum	100%	4920

Risk monitoring by ED

THE UNIVERSITY OF MELBOURNE

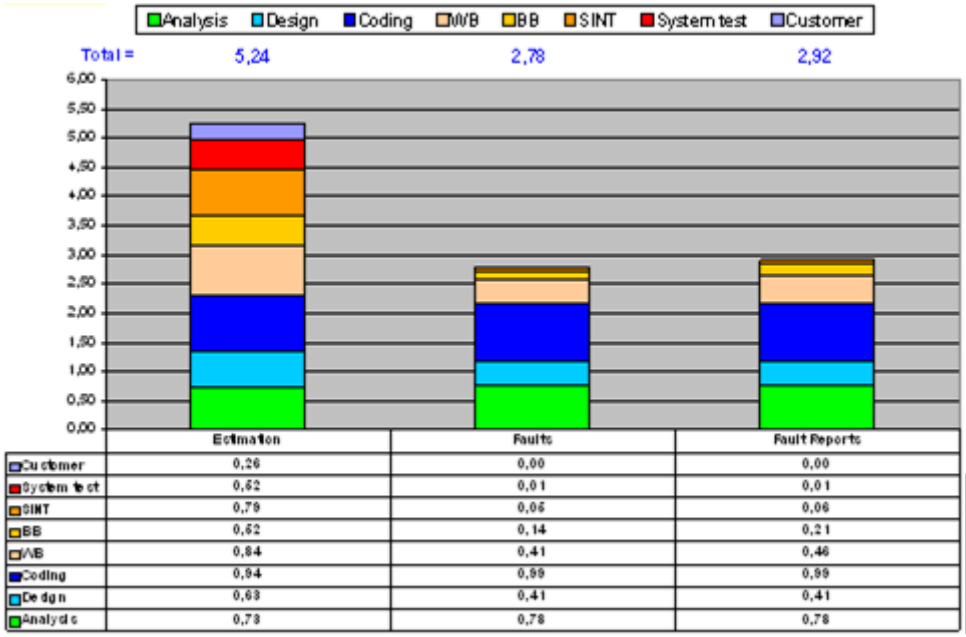
- Estimation against real measures used to monitor the remaining risk in between and at milestones
- What problems can you spot on these 2 charts?



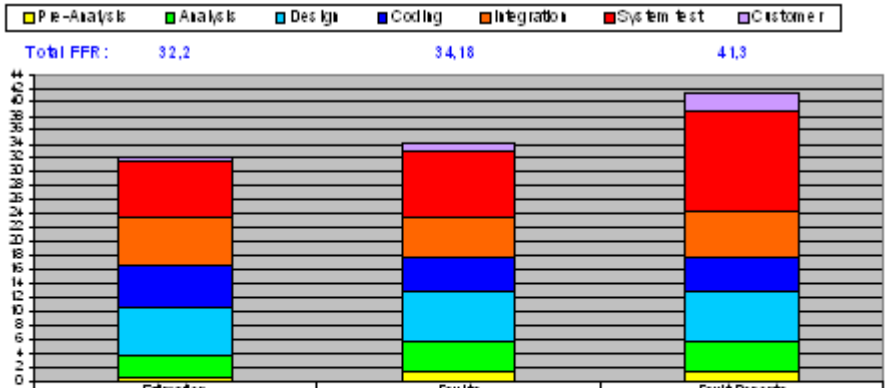


Risk monitoring by ED

- Historical (experimental) data is used to adapt the model to reality for the next project



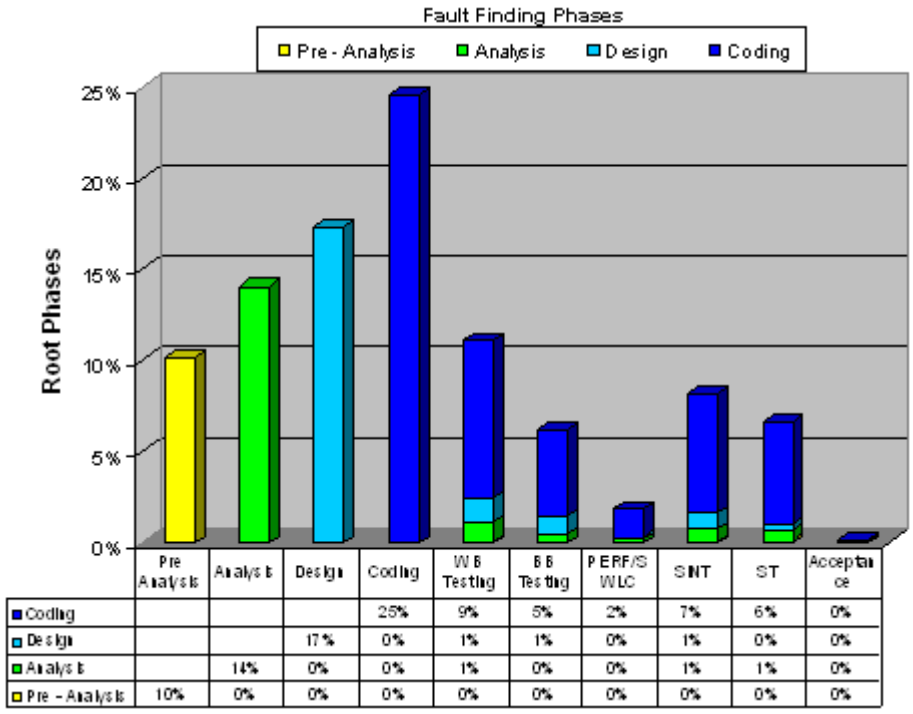
	Estimation	Faults	Fault Reports
Customer	0,26	0,00	0,00
System test	0,62	0,01	0,01
SINT	0,79	0,06	0,06
BB	0,62	0,14	0,21
MVB	0,84	0,41	0,46
Coding	0,94	0,99	0,99
Design	0,63	0,41	0,41
Analysis	0,78	0,78	0,78



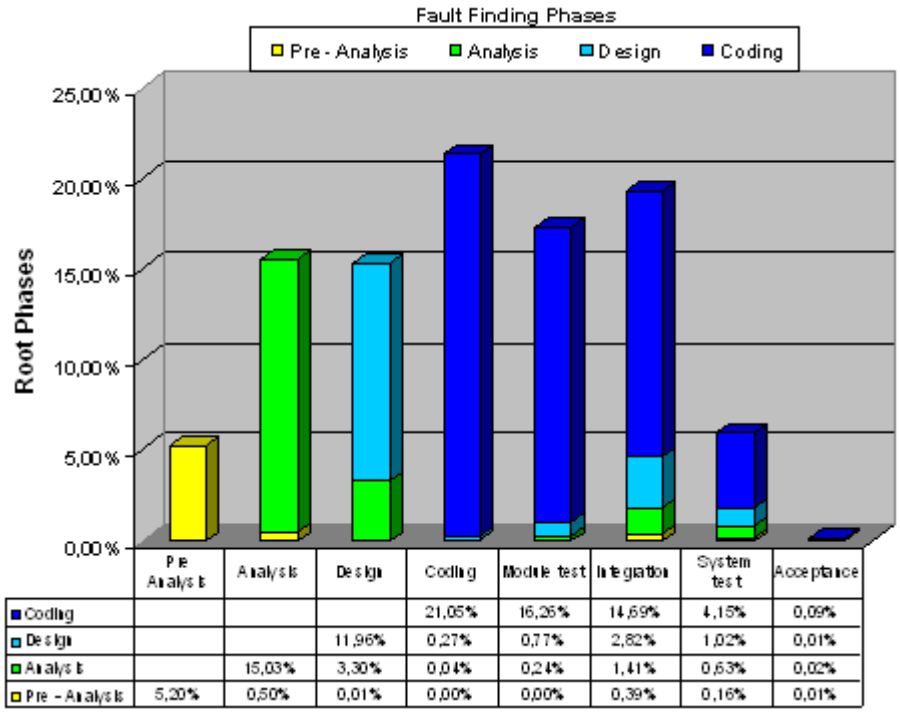
	Estimation	Faults	Fault Reports
Customer	0,7	1,31	2,70
System test	8	9,33	14,42
Integration	7	6,69	8,41
Coding	6	4,98	4,98
Design	7	7,19	7,19
Analysis	3	4,26	4,26
Pre-Analysis	0,5	1,41	1,41

Escape (Root Cause) Analysis

- Shows the defect stream – escape from originating phase to finding phase



	Pre Analysis	Analysis	Design	Coding	WB Testing	BB Testing	PERF/S WLC	SINT	ST	Acceptance
Coding				25%	9%	5%	2%	7%	6%	0%
Design			17%	0%	1%	1%	0%	1%	0%	0%
Analysis		14%	0%	0%	1%	0%	0%	1%	1%	0%
Pre - Analysis	10%	0%	0%	0%	0%	0%	0%	0%	0%	0%



	Pre Analysis	Analysis	Design	Coding	Module test	Integration	System test	Acceptance
Coding				21,05%	16,26%	14,69%	4,15%	0,09%
Design			11,96%	0,27%	0,77%	2,82%	1,02%	0,01%
Analysis		15,03%	3,30%	0,04%	0,24%	1,41%	0,63%	0,02%
Pre - Analysis	5,20%	0,50%	0,01%	0,00%	0,00%	0,39%	0,16%	0,01%

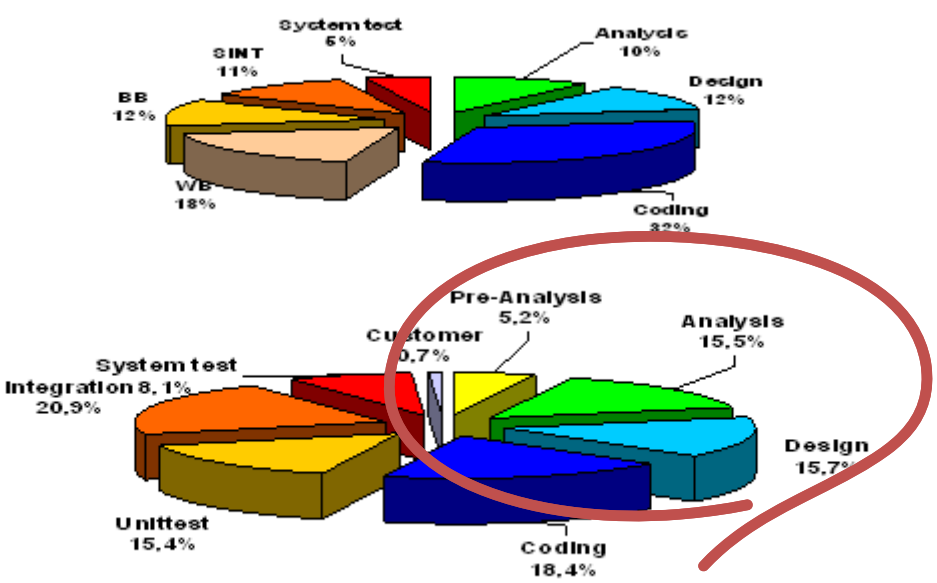
THE UNIVERSITY OF MELBOURNE

SUMMARY – ENTERPRISE

- ✓ Risk – Enterprise Vertical & Horizontal Overview
- ✓ Redesign Crises – Case Study
- ✓ Metrics @ Enterprise Project – Case Study
- ✓ Monitoring, Control & Review – Examples

Significance of Document Metrics

- More than ¼ of all errors to find before Coding
- Majority of errors to find by end of Coding



Error distribution (experimental)	[%]	expected all #errors
Analysis	13.0%	640
Design	13.0%	640
Coding (comp & func test)	30.0%	1476
WBT - FTG	15.0%	738
BBT	9.0%	443
SINT	8.0%	394
ST	8.0%	394
TACC	4.0%	197
SUM	100.0%	4920.0

THE UNIVERSITY OF MELBOURNE

SIEMENS – DOCUMENT METRICS

- ✓ Document metrics
- ✓ SW Code metrics – Code Review
- ✓ Metrics Case Studies – Evaluate, Control & Adapt
- ✓ Conclusion

The Goal to set for the Metrics

THE UNIVERSITY OF MELBOURNE

- Metrics set for the # of errors to find in these 3 phases
- Need processes and metrics
  - to collect the # of faults found
  - to ensure that the review carried out has been adequate
  - to evaluate the results
  - to evaluate the process

- Need a Quality Check to adapt the Quality Control

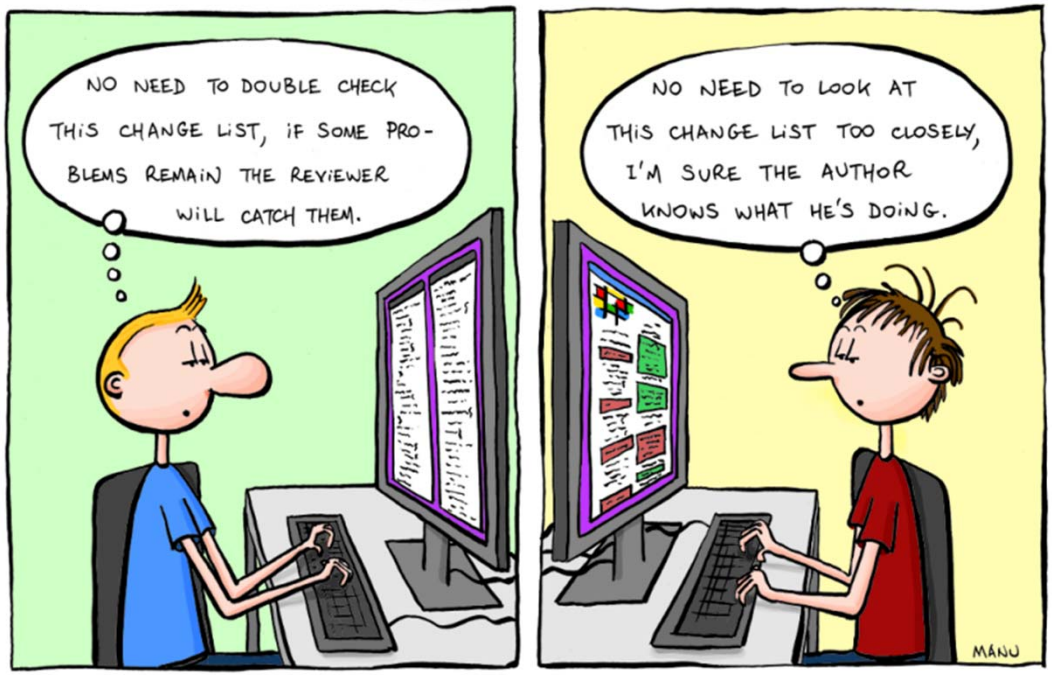
Document Metrics Definitions

THE UNIVERSITY OF MELBOURNE

Review efficiency (RE)

- compares the number of faults with the review effort

$$\mathbf{RE} = \frac{\mathbf{faults}}{\mathbf{review\ effort}}$$



source: http://www.bonkersworld.net/code-reviews/

Document Metrics Definitions

THE UNIVERSITY OF MELBOURNE

Error detection density (EDD)

- number of faults per 100 document pages reviewed

$$EDD = \frac{faults}{pages} * 100$$

- number of faults per Reviewed 1000 Line Of Code (RKLOC)

$$EDD = \frac{faults}{RKLOC}$$

Document Metrics Definitions

THE UNIVERSITY OF MELBOURNE

Review effort density RED

- Review effort per 100 document pages reviewed

$$\mathbf{RED} = \frac{\mathbf{rev.\,eff.}}{\mathbf{pages}} * \mathbf{100}$$

- Review effort per Reviewed 1000 Line Of Code (RKLOC)

$$\mathbf{RED} = \frac{\mathbf{review\ effort}}{\mathbf{RKLOC}}$$

Doc Review - Sample

THE UNIVERSITY OF MELBOURNE

Re-review Required?		No	Reviewer Name:		----		
			Hours spent for Review:		0.5		
Err No.	Error Type	Section/ Page	Review Comment	Status	Escape Phase	Category	Author's Comments
4	TE	2	CI List should be extended with Drop_2, Drop_3 and LSU III artefacts (both baselined and version controlled)	Accepted			Updated CI List.LSU -3 updates are out of scope of this CM Plan
5	TE	2.2.3	Branch conventions for Drop_2, Drop_3 and LSU III missing	Rejected			Branch conventions are applicable only to "Integration" branch ie Kovalam .
6	TE	3.2	LSU III aspects missing (for example in section 3.2 Baseline and Merge Strategy)	Rejected			LSU -3 updates are out of scope of this CM Plan.

- Error Types:
  - RE : Non-conformance with Requirements, missing requirement, incorrect requirements handling
  - TE : Technical error
  - Doc : Documentation error, editorial errors (not counted as errors in the metrics)
  - Remark/hint : suggestions for improvements (not counted as errors in the metrics)

Doc Review – Case Study

Size	
Total no. of pages/KNLOC	49
No. of new/changed pages/KNLOC	49
Time	
Total hours spent for review	3
Total hours spent for rework	2
Error Summary(TE/RE Errors)	
Total number of accepted errors	**3**
Total number of rejected errors	**2**
Total number of open errors	**0**
Total number of duplicate errors	0
Total number of clarified errors	0
Metrics for this report	
RE	1
EDDDR(valid for Docs)	6.12
EDDCI(valid for Code)	0.06

- There were 6 mandatory reviewer
- Total Review Effort is unacceptable
- RED=3/49*100≈6 is far too low
- # of faults found (refer to EDD) – is far too low
- Conclusion: review leaves too high risk in document to progress with, further review has to be carried out

THE UNIVERSITY OF MELBOURNE

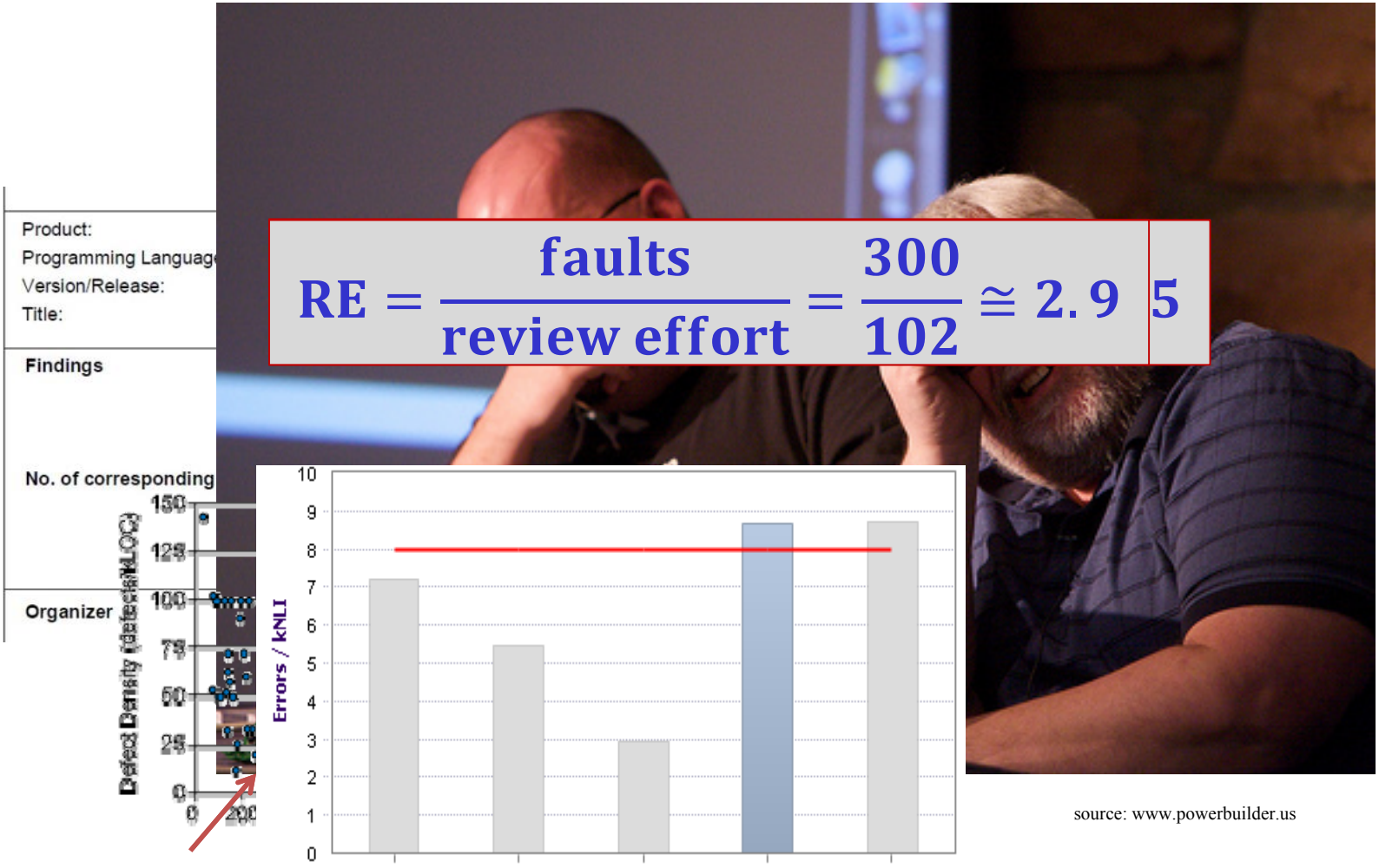
CODE REVIEW

Code Review - sample

THE UNIVERSITY OF MELBOURNE

No.	Line #	Module	Reviewer	Type of Change	Explanation of Problem	Commentary	Acc.
60.	537		BA	TE	the function does not do anything and it is virtual. Could it be = 0 ?(Is it abstract)?		Y
swm_fthngrncdownloadswtask.hpp@@/main/FTH_ngRNC_RC9/3							
61.	19		BA	TE	fwd decl seems to be unnecessary		Y
62.	34		BA	TE	operator= has no definition	operator= isn't used	N
63.	[illegible]		BA	TE	destructor should be virtual	This class has no descendants.	[illegible]
64.	[illegible] 730		BA	TE	I know that it is a quasi necessity here to use void* but rule9.20 sais "void*" is strictly forbidden	Yes it is, but SYS_Corbatimer was designed with a generic pointer, so that's why we need this	N
swm_fthngrncdownloadswtask.cpp@@/main/FTH_ngRNC_RC9/3							
65.	Gen.		BA	TE	terminate macro calls with semicolon		Y
66.	Gen.		BA	TE	better to use '!' instead of "if (somePtr==0)"		Y
67.	87		BA	TE	The function has a TRC_ENTRY but no TRC_EXIT here.		Y
68.	90 208 343 419 496 727		BA	TE	return is superfluous here	I like returns in the end of the functions and it costs nothing ☺	N
69.	[illegible]		BA	TE	use static_cast<>		Y
70.	164		BA	TE	rule9.20 sais "void* is strictly forbidden	See comment No 64	[illegible]
71.	172		BA	TE	I'm not wrong SYS_ERROR_FILL writes trace automatically so this		Y

Code Review – Case Study



$$\mathbf{RE} = \frac{\mathbf{faults}}{\mathbf{review\ effort}} = \frac{\mathbf{300}}{\mathbf{102}} \cong \mathbf{2.9}$$

5

source: www.powerbuilder.us

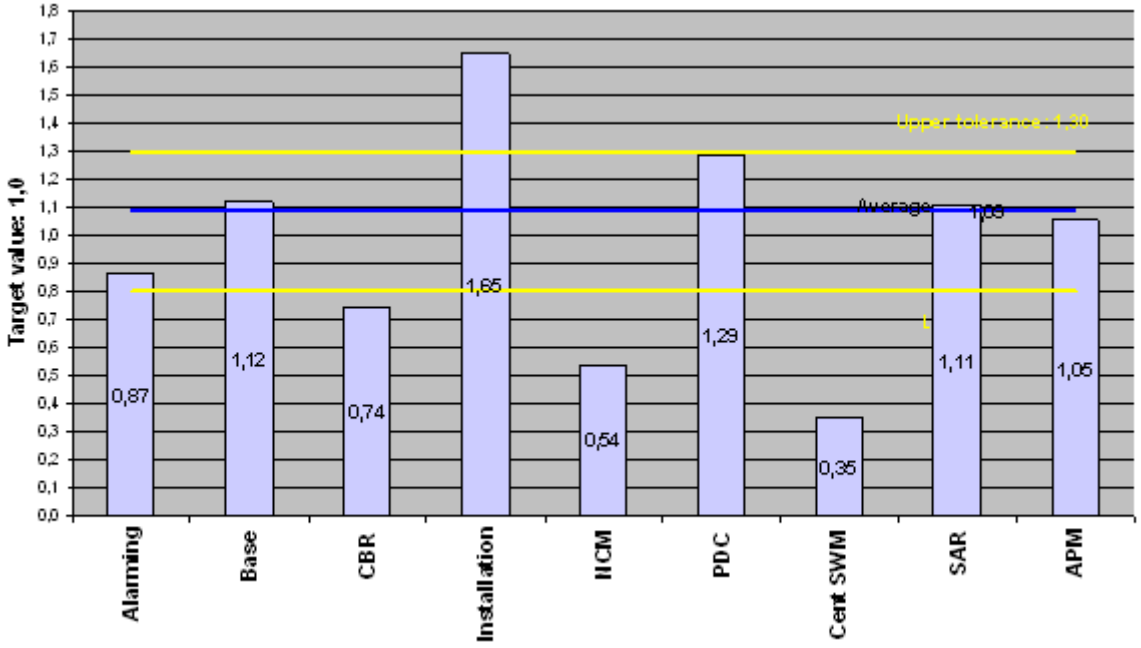
Further step on code review: http://www.ibm.com/developerworks/rational/library/11-proven-practices-for-peer-review/index.html

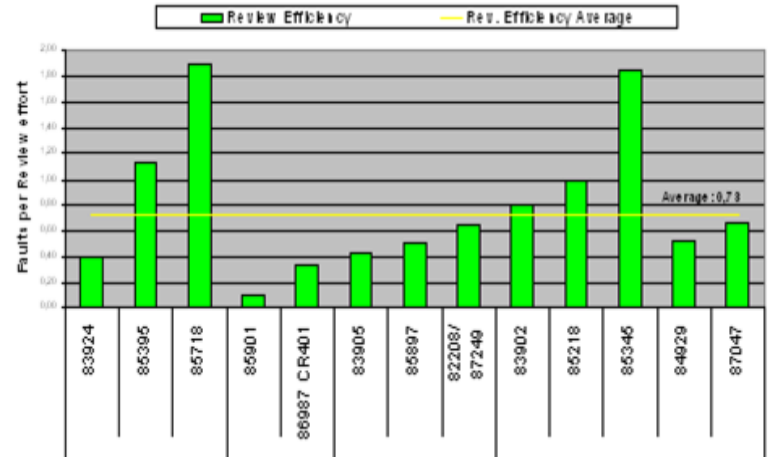
THE UNIVERSITY OF MELBOURNE

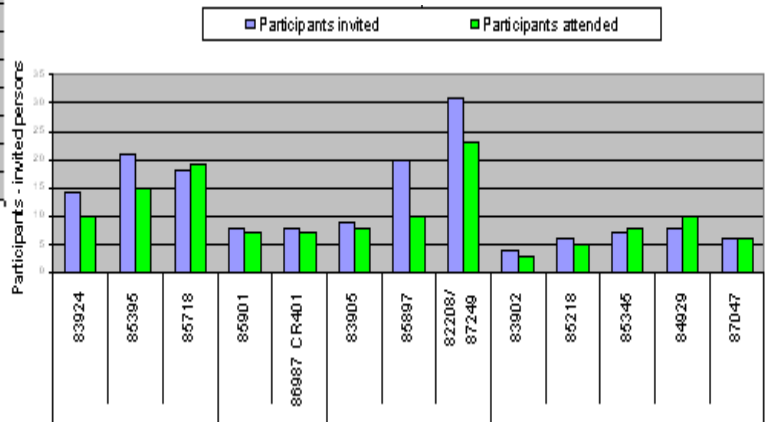
METRICS CASE STUDIES

RE example (faults/effort)

- Efficiency Control – used to spot risk at an early stage

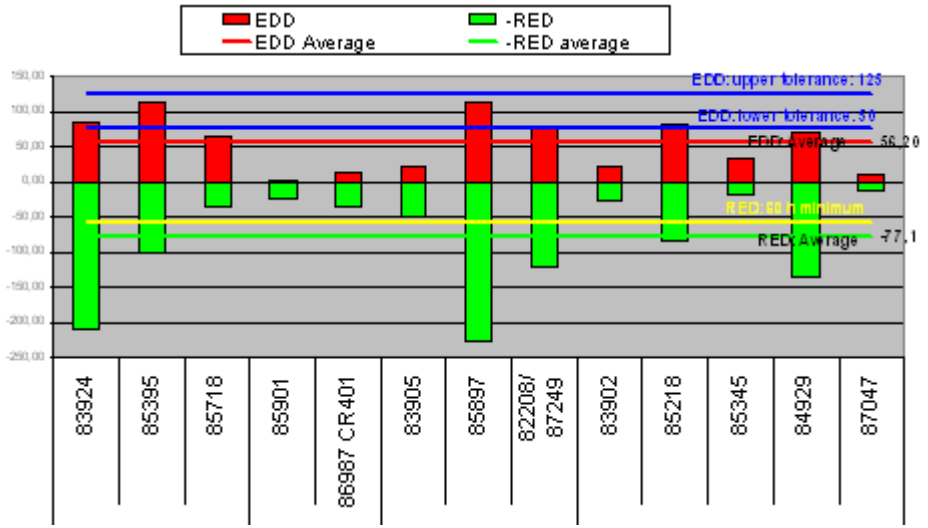




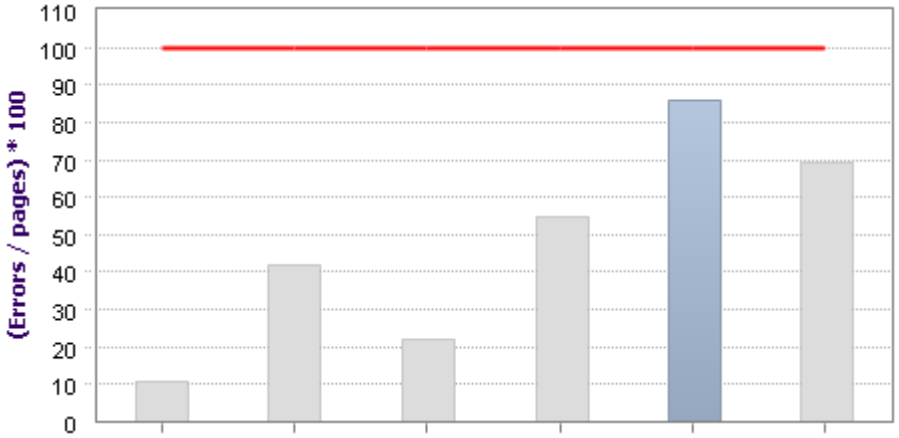


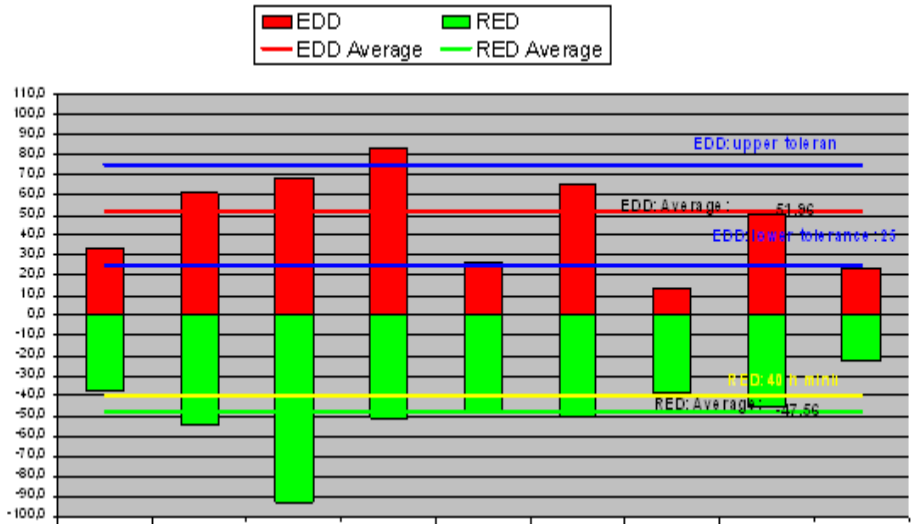
EDD & RED – doc review

- EDD – faults/volume checked
- RED – effort/volume checked
- Quality Control of the Quality Check
- Manage the risk associated with poor Quality Control on execution



Feature Sheet Reviews/NEys





Data Quality - Analysis

THE UNIVERSITY OF MELBOURNE

Review results have to be analyzed by RQM (Risk & Quality Manager)

Goal

- ❑ **Quality** Status **of** phase **results**
- ❑ **Quality** Status **of development** phase
- ❑ **Identify** weak points and problems
- ❑ Make proposals for **improvement**

source: www.dashboardinsight.com/articles/new-concepts-in-business-intelligence/data-analysis-overview.aspx

Checklist

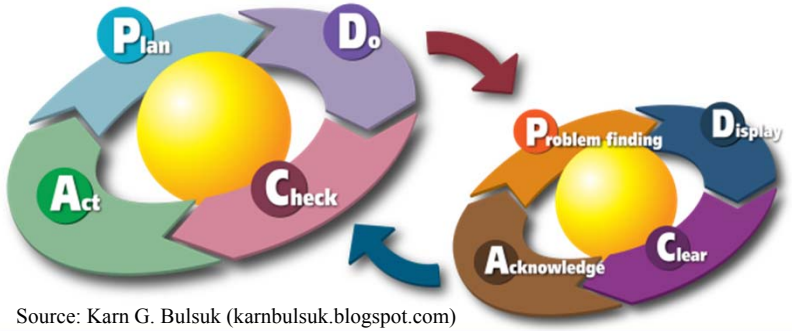
- ❑ Check review records (samples or all)
- ❑ Is spend **Review Effort** adequate?
- ❑ **Fault Finding** is ok?
  (too high, too low)
- ❑ Definition and usage of metrics to compare with former results
- ❑ Compare current w estimated values
- ❑ Definition and monitoring of improvement measures
  (if deviations from estimated targets or quality problems are recognized)

THE UNIVERSITY OF MELBOURNE

CONCLUSION

Conclusion

- Real World requires **adaption**
- Depth of Risk Management and Quality Control is in contrast for a small company and for an enterprize
- **Continuous** Risk & Quality Management
- Monitor the Risk & Eliminate the **Waste**
- Look for the **Best Practice** in the industry



Source: Karn G. Bulsuk (karnbulsuk.blogspot.com)

From Risk to Best Practice

THE UNIVERSITY OF MELBOURNE

- **Risk Management**
  – mistakes and crises leads **to better practice**
- **Best Practice** is from **Crises Management**
- **Crises Management** leads to
  Adaptive Project Management
- **Adaptive Project Management** is an
  Active Risk Management

Agile Alliance

CERTIFIED Scrum Alliance Scrum Product Owner

CERTIFIED Scrum Alliance ScrumMaster

Lean Six Sigma

THE UNIVERSITY OF MELBOURNE

SUMMARY – DOCUMENT METRICS

- ✓ Document metrics
- ✓ SW Code metrics – Code Review
- ✓ Metrics Case Studies – Evaluate, Control & Adapt
- ✓ Conclusion