

Risk based Testing

Andreas Golze

Global Practice Management



Agenda

- ▶ Motivation for Risk based Testing
- ▶ Definitions
- ▶ Maturity levels of Risk assessment
- ▶ Example
- ▶ Summary

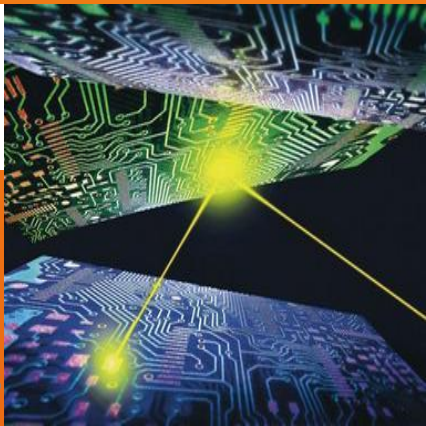
Motivation for Risk based testing



Motivation for Risk based testing

- Divide the application into comparable usable elements which have from an end user perspective an impact on the business
- Get objective criteria for the decision which test method should be applied for each of the elements

Definitions



Definition for Risk

- A risk in software threatens one or more of the function the software is expected to perform and has a uncertain probability.

- Risk is normally calculated:

Risk = damage cause by failure X

probability of the failure to happen

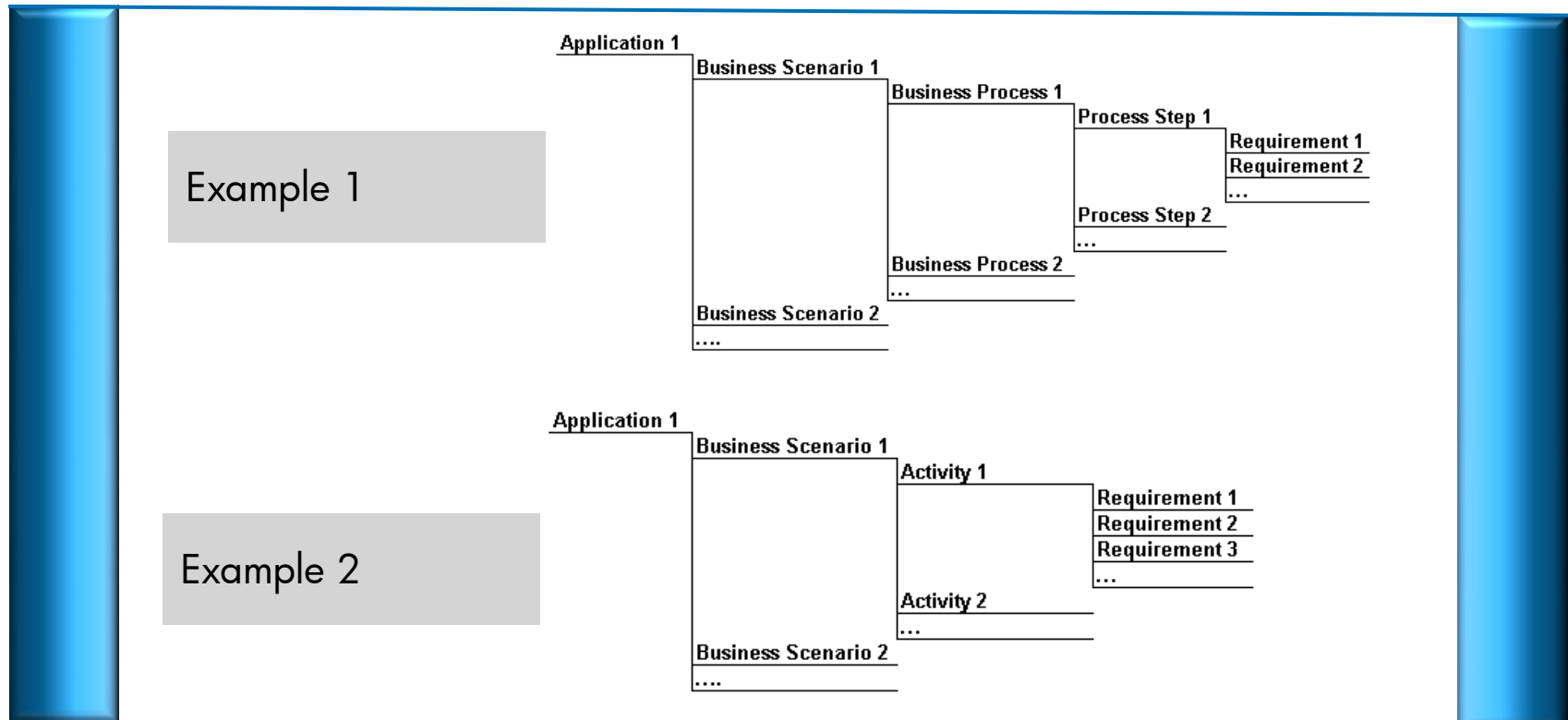
IMPORTANT:

Risk exists only where there is uncertainty the formula gives only a indication about the magnitude of the risk does not solve the uncertainty

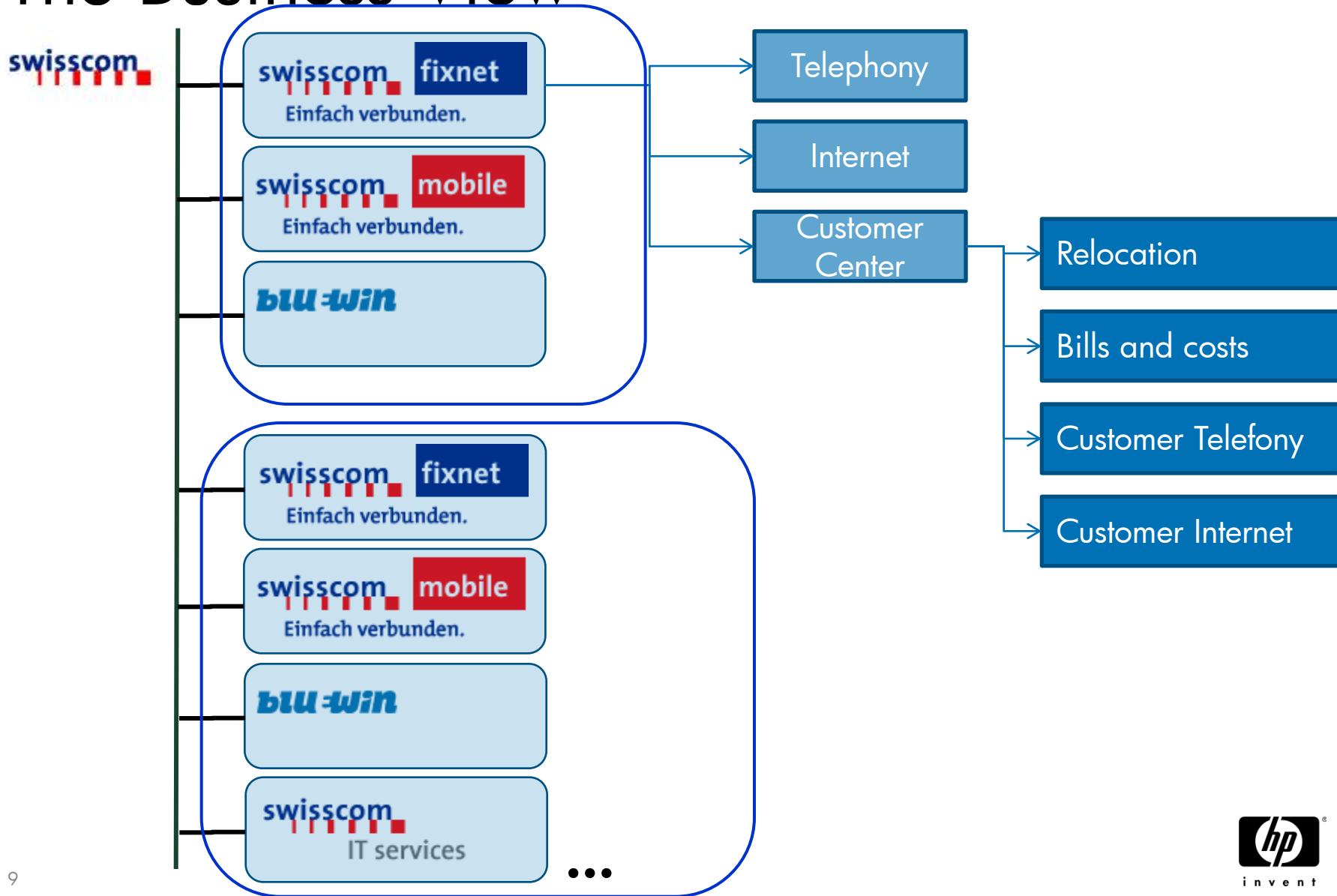
Requirements Structure

- Organization of Requirements

- Applications support business activities/processes and functions
- Structure is uniform and enforced, and naming convention is informative



The Business View



Business Impact Driven Test Strategy

Analyzing Business Impact

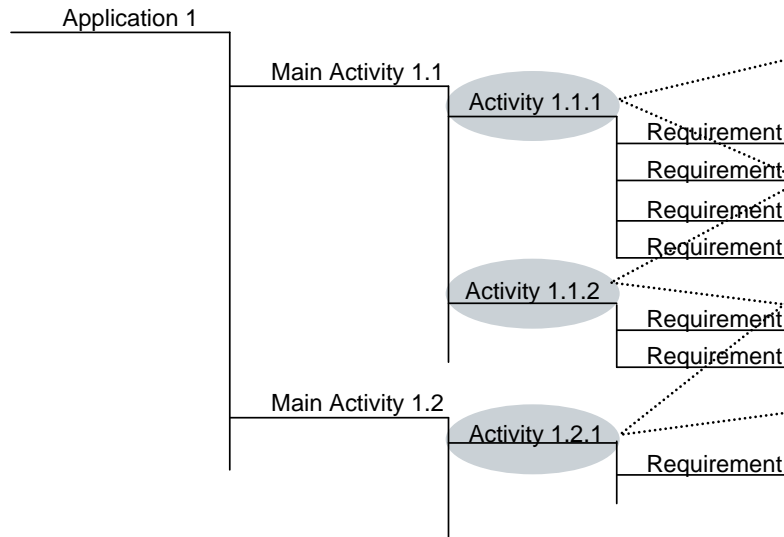
Result	A	B	C
Criteria	High Risk	Medium Risk	Low Risk
Type of Process	Calculation / validation	change of data	display
Business Impact	legal	wrong information	none
Frequence of use	very often	often	rare
Number of Customer affected	large number / very important	group	some

Analyzing Probability

Result	III	II	I
Criteria	Unlikely	Possible	Likely
Change Rate	Unchanged	Changed Func.	New Func
Platform OS	HOST	UNIX or Windows	Unix, Windows, Host combination
Defects Rate	Low	Medium	High

Determining the Risk

Prob.	III	II	I
Impact	Un-Likely	Possible	Likely
A	B	A	A
B	C	B	A
C	C	C	A

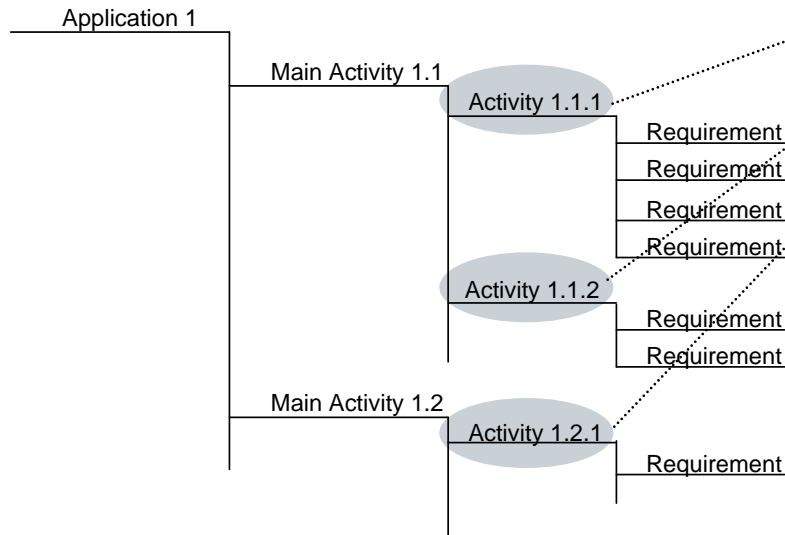


- Assess the business impact of each activity
- Assess the Probability for failure
- Determine the Risk based on Impact and probability

The Risk determines the Test Strategy

Procedure for High Risk A	Procedure	Systematic testing using business component testing strategy, and root-cause analysis
	Approach	Automated: 30% Manual: 70%
Procedure for Medium Risk B	Procedure	Systematic testing with or without business component testing strategy, and root-cause analysis
	Approach	Automated: 20% Manual: 80%
Procedure for Low Risk C	Procedure	Systematic or Ad-hoc testing
	Approach	Automated: 5% Manual: 95%

Functional Complexity determines Test effort



Analyzing Functional Complexity

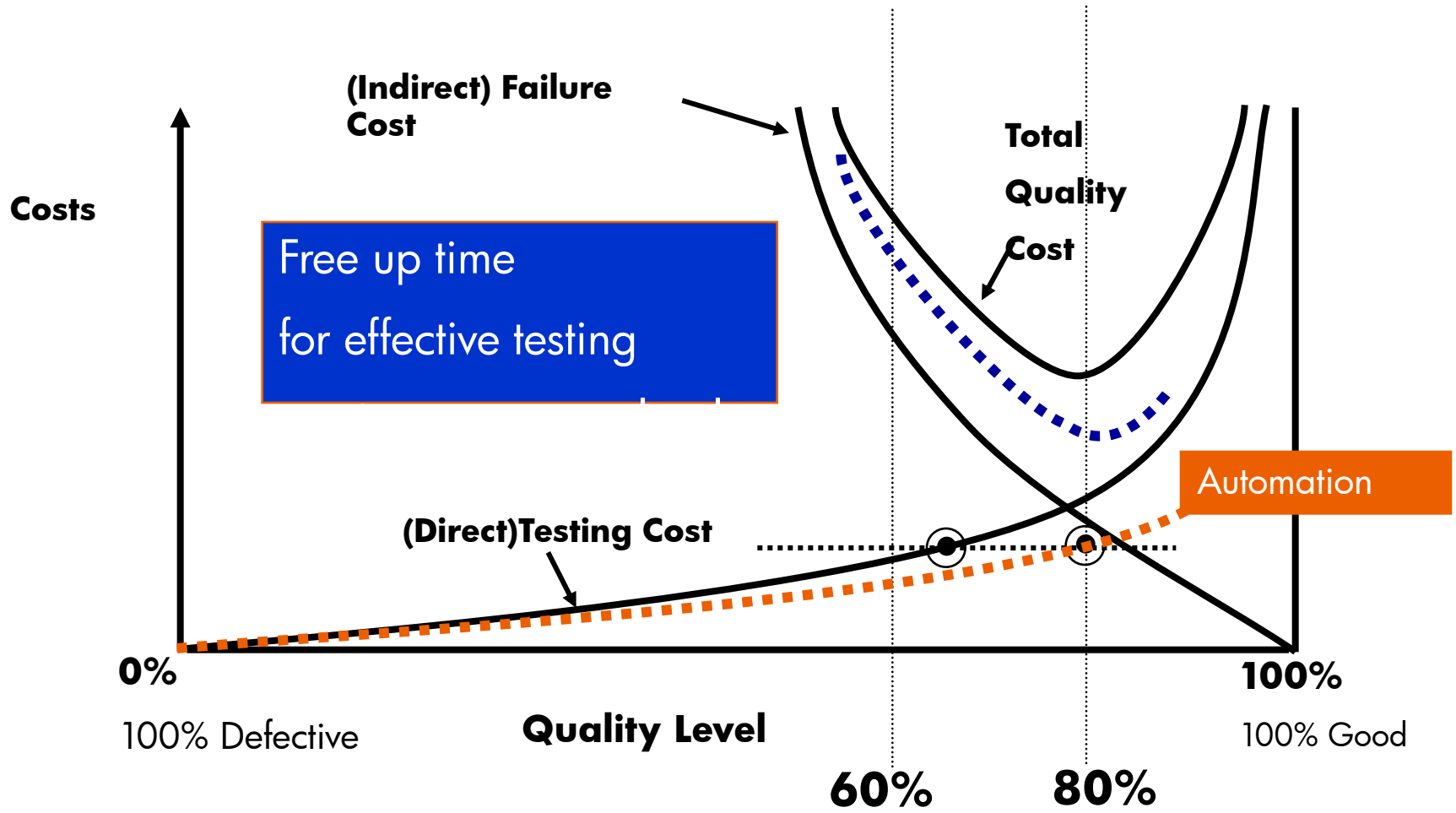
Complexity	1	2	3
Criteria	High complex	Medium complex	Low complex
Number of Objects affected	> 9	4 - 9	< 4
Number of Objects with write access	> 3	1 - 3	< 1
Number of objects with read access	> 5	3 - 5	< 3
Number of Windows affected	> 4	2 - 4	< 2

Determine Testing Effort

Complexity	1	2	3
Impact	High complex	Medium complex	Low complex
A Critical	12,5	10,2	8,3
B Important	6,9	6,3	5,6
C Nice to have	4,2	3,5	2,8

- Analyse the functional complexity together with a Technical Architect
- Determine the Test Effort by combining the Risk and the Complexity

Automation as Part of The Strategy



Source: J.M. Juran's Quality Control Handbook
Giga Information Group 2001, Justifying IT Investments: Quality Assurance

Business Impact Matrix

Result Criteria	A High Impact	B Medium Impact	C Low Impact
Type of Process	Calculation / Validation	Change of data	Display
Business Impact	Legal	Wrong information	none
Frenquence of use	Very often	Often	Rare
Number of Customers affected	Large number / Very important	Group	Some

The defect rate refers to either the past experience/ defect in production or to defects detected along the project.

This might add a dynamic effect to risk which makes it more interesting – “project intelligence”

Probability Matrix

Result Criteria	III Unlikely	II Possible	I Likely
Change Rate	Unchanged	Changed Function	New Function
Software maturity	Mature (> 10 years)	Progressing (5 – 10 years)	Immature (< 5 years)
Defects Rate	Low	Medium	High

The defect rate refers to either the past experience/ defect in production or to defects detected along the project.

This might add a dynamic effect to risk which makes it more interesting – “project intelligence”

Risk Determination

Prob. / Impact	III Un-Likely	II Possible	I Likely
a	B	A	A
b	C	B	A
c	C	C	B

- The Risk is a product of the Impact and the Probability
- Both values should be calculated and kept separately (for each Business Activity)
- The user should be able to override the calculated Impact and Probability
- The effort allocation should be done referring to the above Risk Matrix

Functional Complexity Matrix

Complexity Criteria	1 High complex	2 Medium complex	3 Low complex
Number of Objects affected	> 9	4 - 9	< 4
Number of Objects with write access	> 3	1 - 3	< 1
Number of objects with read access	> 5	3 - 5	< 3
Number of Windows affected	> 4	2 - 4	< 2

Test Procedures –

Approach based on Risk

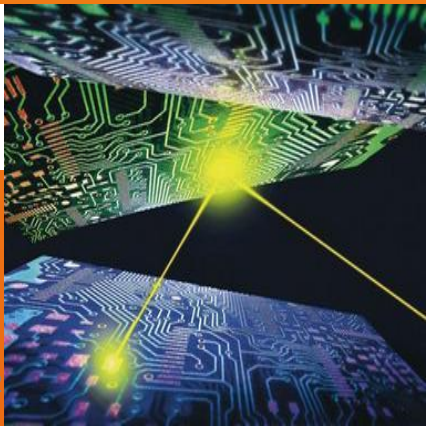
Procedure for High Risk	Procedure	Systematic testing using business component testing strategy, and root-cause analysis
	Approach	Automated: 30% Manual: 70%
Procedure for Medium Risk	Procedure	Systematic testing with or without business component testing strategy, and root-cause analysis
	Approach	Automated: 20% Manual: 80%
Procedure for Low Risk	Procedure	Systematic or Ad-hoc testing
	Approach	Automated: 5% Manual: 95%

Test Effort

Complexity Risk	1 High complexity	2 Medium complexity	3 Low complexity	Test Procedure
A High Risk	13 - 19	10 - 16	7- 13	Procedure for <u>high</u> Risk
B Medium Risk	6,5 - 9,5	6,5 - 9,5	4 - 6,5	Procedure for <u>medium</u> Risk
C Low Risk	2,3	2,2	1,2 - 2,2	Procedure for <u>low</u> Risk

Effort in PD's for Test Case development, Test Data definition, Test Script development, Test execution and Test evaluation per Process Step

Maturity levels of Risk Assessment



Capability Levels (CMMI)

- **1 Initial**
- **2 Managed**
- **3 Defined**
- **4 Quantitatively Managed**
- **5 Optimizing**

Level 1: Initial (heroic efforts)

- Risk is assessed based on the experience of the tester. There is no formal process and each tester decides how to test based on his experience

Level 2: Managed

- Risk is assessed based on the Business Impact Matrix. Each project is using the matrix in their way

Level 3: Defined

- Risk is assessed based on the Business Impact Matrix, and the Probability Matrix. The usage of both is mandatory and enforced with a tool (QC). The rules for the customization of the matrices are defined and documented.

Level 4: Quantitatively Managed

- Risk is assessed based on the Business Impact Matrix, and the Probability Matrix. The usage of both is mandatory and enforced with a tool (QC). The Business Impact Matrix is extended with values for damage the Probability is calculated on likelihood values (RUM)

Q&A

