

Smartest Technologies Ltd.

Providing the SMART solutions

How to find more bugs? (Especially if you have no time)

11th SQA Days & ELT Ukraine

April 2012

Yaron Tsubery

Website: www.dSmartest.com

Email: YaronTs@dSmartest.Com

Mobile: +972 52 8549255



Agenda

Introduction

Presentation objectives

The challenges that we have

Take control over the process

Summary





Introduction

- ❖ Yaron Tsubery
- ❖ Homeland: Israel
- ❖ More than 19 years experience in Software Development and Testing,
- ❖ Director QA & Testing at Comverse, managed large testing groups and projects deployed to customers located in Israel, USA, Europe, and to the Fareast countries and...
- ❖ Currently, Managing Director of Smartest.
- ❖ President of ITCB (Israeli Testing Certification Board) and a formal member in ISTQB,
- ❖ President of ISTQB (International Software Testing Qualifications Board).

Agenda

Introduction

Presentation objectives

The challenges that we have

Take control over the process

Summary





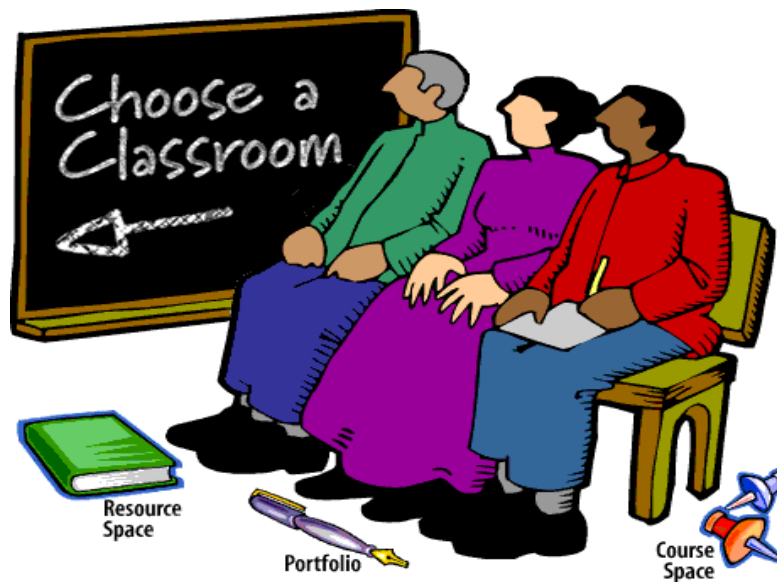
Objectives

- ❖ We will walkthrough some theoretical material and see the **match between theory and reality**,
- ❖ Present the **challenges** and their **results**,
- ❖ Show a way to **improve** your **efficiency and effectiveness**, through a **mixed Approaches**,
- ❖ The presentation is aimed at **stimulating your mind** and opening new views to the subject,



Set Expectations

- ❖ This presentation focuses on the testing process related to the **Execution phase**,
- ❖ Let's have a **dynamic and interactive** session



Agenda

Introduction

Presentation objectives

The challenges that we have

Take control over the process

Summary





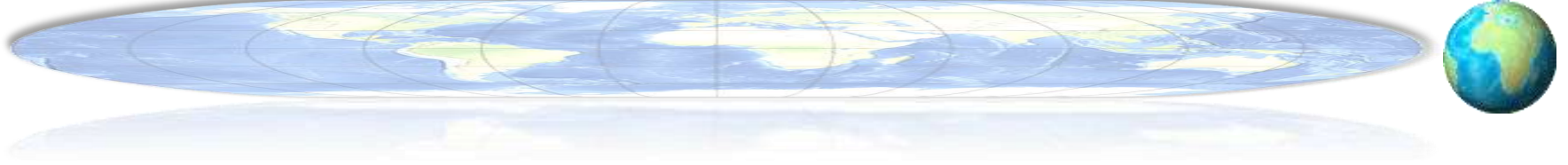
What Were The Challenges That We Had?



Challenges

- ❖ Very **large** and **complex** systems
- ❖ Systems that **required** to function **24/7**
- ❖ **Frequent** requests for **changes**
- ❖ **Market** change and required **fast delivery**
- ❖ Continue to **keep the high quality level** required for our systems
- ❖ **Competitors...**





Background

Life-cycles Models (SDLCs)

❖ Big Bang

❖ Water fall

❖ Spiral

❖ V-Model

❖ Time-box and Agile development model

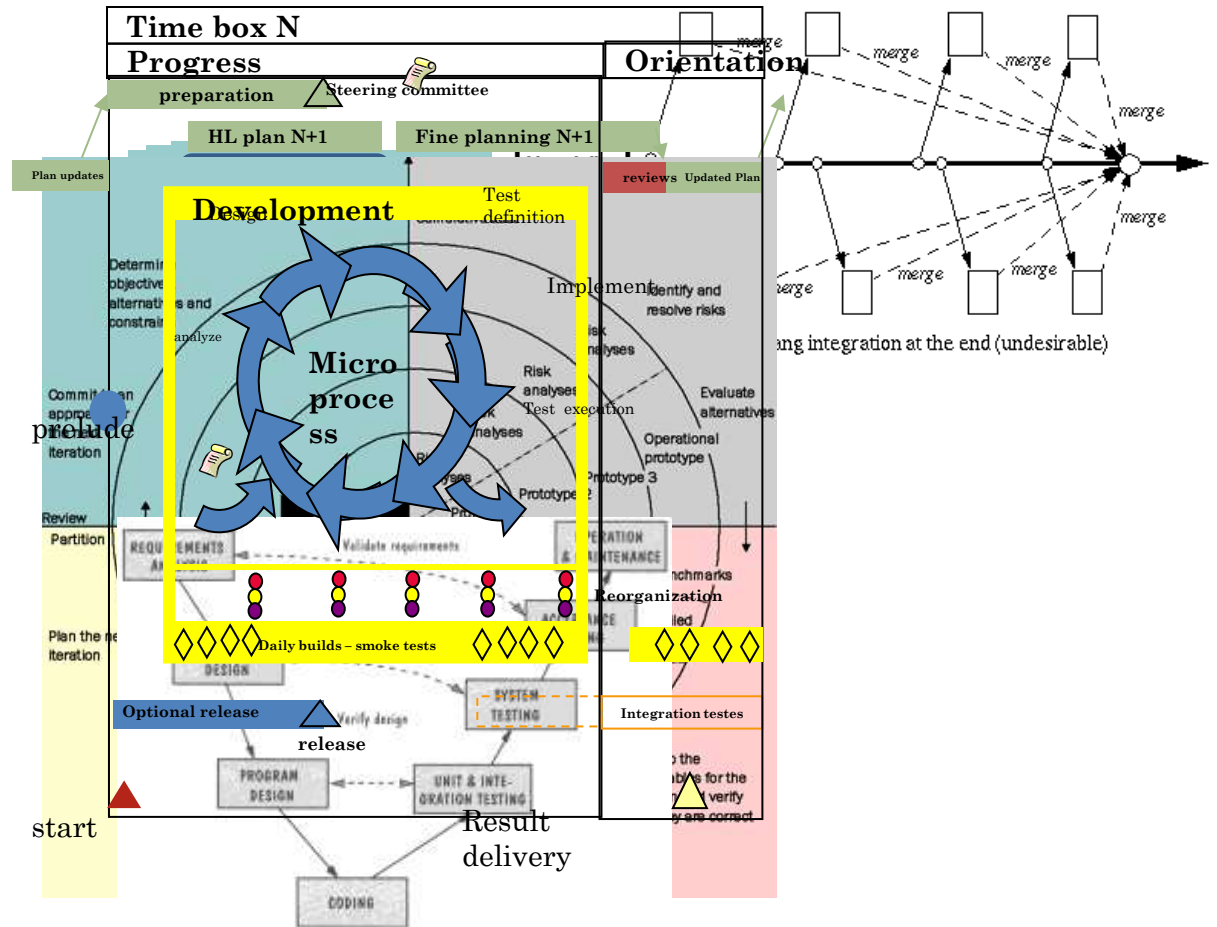
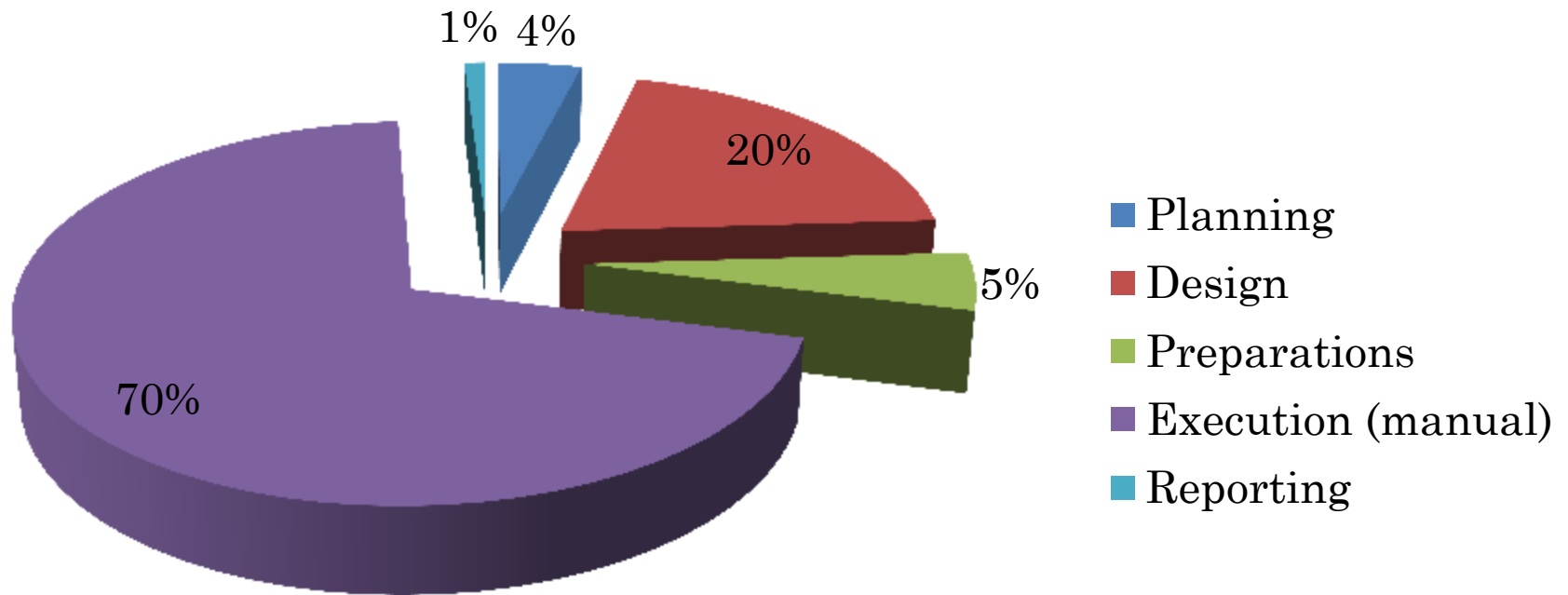


FIGURE 2.4 The V model.

Were Usually We Put The Efforts In The Testing Process?

How much time are you investing in...?

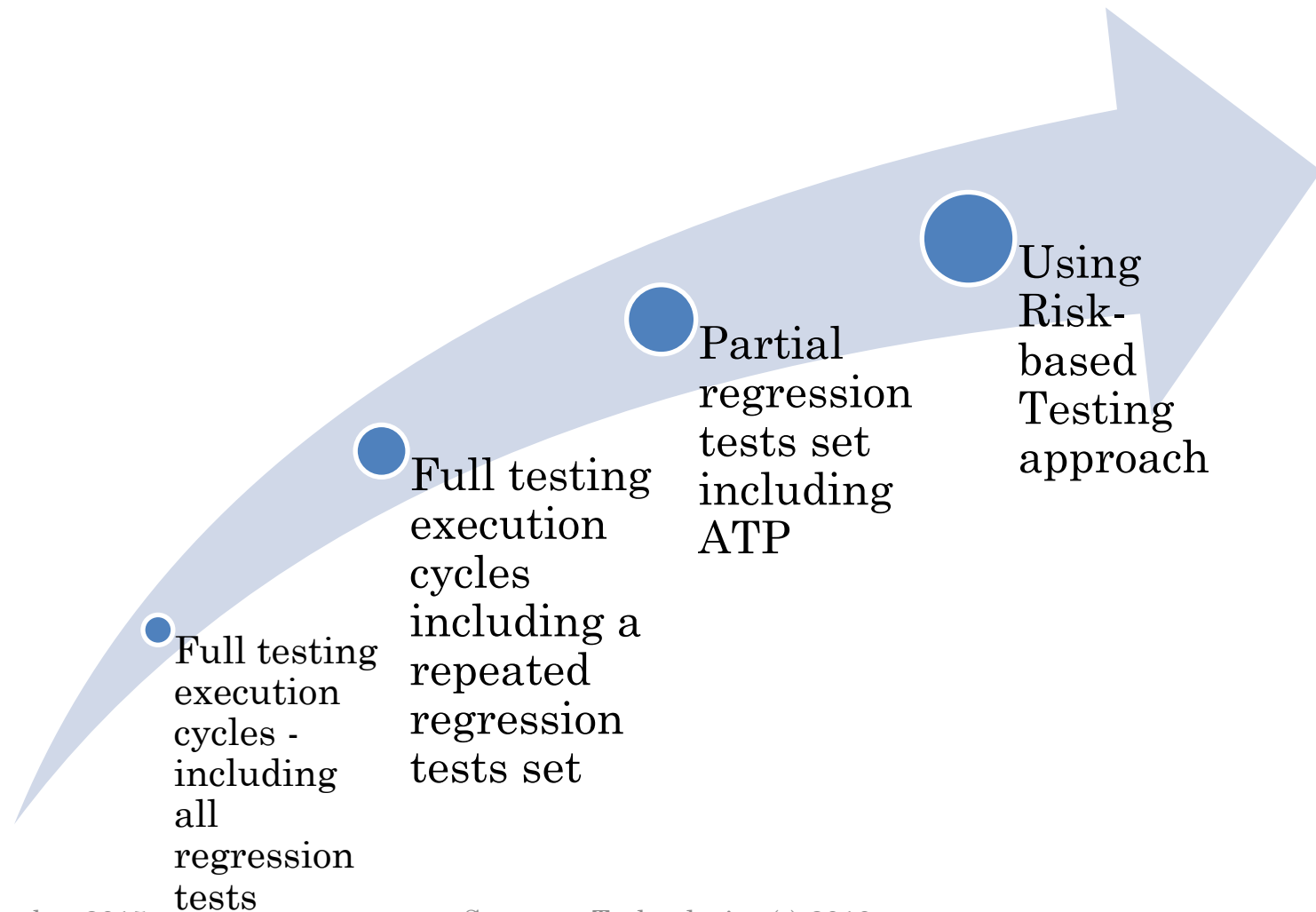




Where Do We Invest Most of Our Time at The Execution Stage?



The Evolution Of Regression

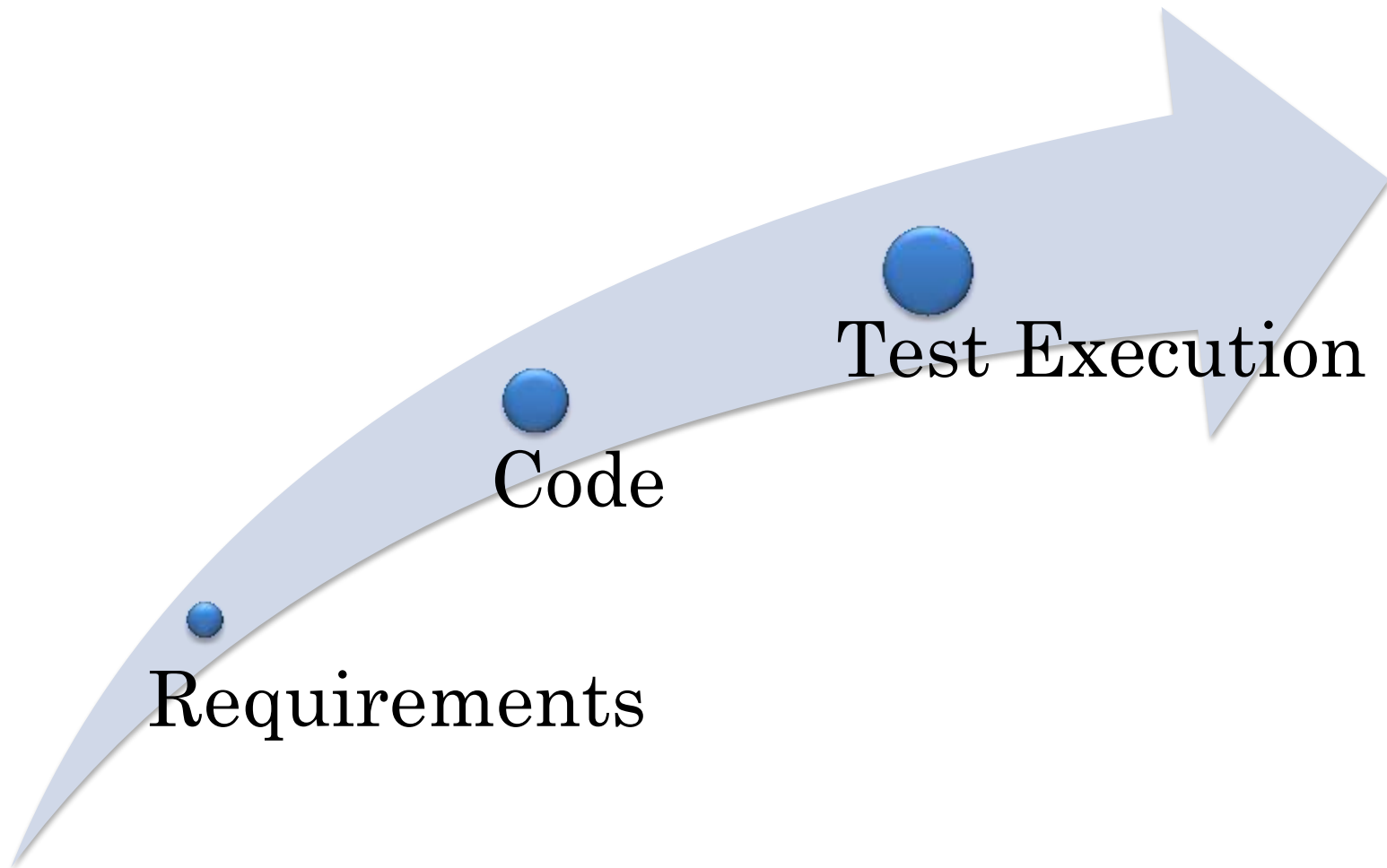




What Are We Looking For?



Stages That We Can Improve Defect Detection





Nothing New So Far... (-;

Agenda

Introduction

Presentation objectives

The challenges that we have

Take control over the process

Summary






So, What Did We Do? How Our Strategy & Plan Looked Like?



Strategy

- ❖ **Focused plan at the Execution Cycles**
 - ❖ Use **risk-based** approach
 - ❖ Rapidly **track** and **analyze** the **execution results**
 - ❖ **Manage** through **defect detection** analysis.
- 



Planning

- ❖ Main dimensions to consider at execution:
 - Concentrate on **each build** we're getting from the development
 - **Design** tests using **quality risk categories** analysis
 - Divide each build to **3 parts** – following **risk-based** approach, using **priority** (High, Medium and Low)
 - **What** to cover in each part?
 - Which **coverage level** should we have?
- ❖ How can we **benefit** from **previous builds** and cycles?
- ❖ How can we **benefit** from the **current build**?
- ❖ ...and even from the **current part** we're testing!





Level Of Testing

- ❖ Risk analysis and **priorities** of the features – taken from the **initial plan** we built
- ❖ Evaluation of level according to **sanity tests results**
- ❖ **Dependencies** between features (new & current)
- ❖ Coverage we had in **previous build**:
 - **Did** we **cover** that part?
 - **How much** and what did we cover?
 - What were the **results** of previous tests?
 - **How many defects** did we had? Their **severity** and **priority!**



Level Of Testing – cont'

❖ Don't forget the Platform and Infrastructure areas:

- New hardware
- New Kernel
- O/S changes
- Changes or New 3rd party parts
- Configurations
- Etc'



❖ And additional parts that fits to your system

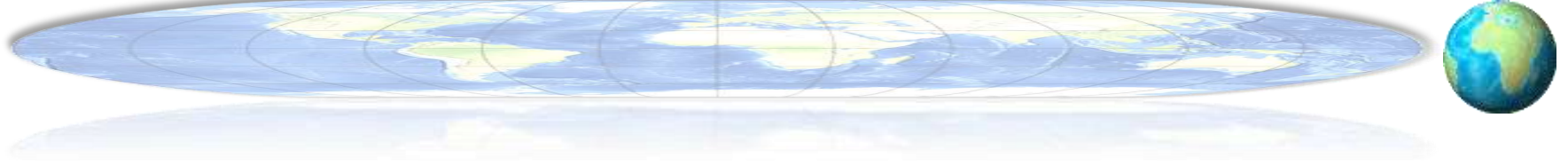
❖ I'm sure that you have now **concerns** about the **duration** which these activities may take...



Coverage Areas

- ❖ Define **what to cover** per each set of priority (High, Medium and Low)

- ❖ Sanity
- ❖ New functionality
- ❖ Infrastructure & system related tests
- ❖ Regressions:
 - Related to new functionality
 - Bug fixes / retests
 - Related to bug fixes
- ❖ ATP related tests



Adding The Focused Exploratory Part



Focused Exploratory Testing

- ❖ After we have the **plan** per each **set of priority** test, considering all **elements of coverage** and level of **coverage**,
- ❖ Add to each set of priority test – **at the end** – a Focused Exploratory Testing part
- ❖ Let's have a look at the Gantt



Gantt Sample

		Task Name	Duration	Start	Finish
27		<input checked="" type="checkbox"/> Test Planning	130 days	Wed 11/1/06	Sun 4/29/07
31		<input checked="" type="checkbox"/> Test Design and preparations	75 days?	Thu 3/1/07	Sun 6/17/07
74		<input type="checkbox"/> Test Execution	118 days?	Sun 5/27/07	Tue 11/13/07
75		<input checked="" type="checkbox"/> STF1 (new content)	28 days?	Sun 5/27/07	Tue 7/3/07
86		<input type="checkbox"/> STF 2 (new content)	27 days?	Sun 6/17/07	Mon 7/23/07
87		<input checked="" type="checkbox"/> General test operation	27 days?	Sun 6/17/07	Mon 7/23/07
93		<input type="checkbox"/> Single Lab - Critical areas	16 days	Tue 6/26/07	Tue 7/17/07
94		KIT/Installation tests	2 days	Tue 6/26/07	Wed 6/27/07
95		Sanity tests	1 day	Thu 6/28/07	Thu 6/28/07
96		CSM - progression tests	10 days	Sun 7/1/07	Thu 7/12/07
97		Retest defects	5 days	Sun 7/1/07	Thu 7/5/07
98		Regression tests	5 days	Sun 7/8/07	Thu 7/12/07
99		Analysis & Exploratory tests	3 days	Sun 7/15/07	Tue 7/17/07
100		<input type="checkbox"/> LAL	22 days	Sun 6/17/07	Mon 7/16/07
101		<input type="checkbox"/> Integration tests	4 days	Sun 6/17/07	Wed 6/20/07
102		E2E Flows	4 days	Sun 6/17/07	Wed 6/20/07
103		<input type="checkbox"/> Build tests	15 days	Tue 6/26/07	Mon 7/16/07
104		<input type="checkbox"/> Critical Areas	14 days	Tue 6/26/07	Sun 7/15/07
105		Sanity tests	1 day	Tue 6/26/07	Tue 6/26/07
106		E2E Flows	6 days	Wed 6/27/07	Wed 7/4/07
107		Load tests	5 days	Thu 7/5/07	Wed 7/11/07
108		Analysis & Exploratory	2 days	Thu 7/12/07	Sun 7/15/07
109		<input type="checkbox"/> Other areas	5 days	Tue 7/10/07	Mon 7/16/07
110		HA test	3 days	Tue 7/10/07	Thu 7/12/07
111		Monitoring	2 days	Tue 7/10/07	Wed 7/11/07
112		Security and system procedures	3 days	Tue 7/10/07	Thu 7/12/07
113		Analysis & Exploratory	2 days	Sun 7/15/07	Mon 7/16/07
114		<input type="checkbox"/> STF 3 (new content)	30 days?	Mon 7/23/07	Sun 9/9/07



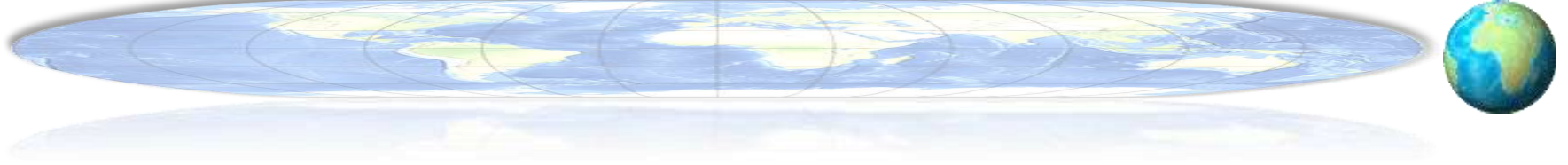
Well...

**How To Leverage
The Efficiency of
Exploratory
Testing?**



Leveraging The Exp' Testing

- ❖ Perform an **analysis** of the **defects density**
 - Severity
 - Priority
- ❖ Don't delay the **analysis** to the end of the testing cycle, perform that **while you execute** the test priority set!
- ❖ **After analyzing** the defects density and the infected areas, chose the **most infected areas** and focus your exploratory testing there!
- ❖ Now you're **running** focused **exploratory testing**



Let's Simplify The Defect Detection Analysis?

Defects Density Per Area

❖ Assumptions:

- We have 2 new functionality areas: A & B
- We have 2 areas for regressions: L & M

❖ We saw the following defect density behavior

Area	Test Group	Severity (Critical)	Severity (Major)	Severity (Minor)	Total (C+Mj)	Total (Mj+Mn)	Total
A	A.1			3	0	3	3
A	A.2			5	0	5	5
A	A.3		1	8	1	9	9
A	A.4			7	0	7	7
A	A.5		1	8	1	9	9
B	B.1		1	1	1	2	2
B	B.2	1	2	3	3	5	6
B	B.3		1	2	1	3	3
L	L.1	1	2		3	2	3
L	L.2	2	3	4	5	7	9
L	L.3	1	2		3	2	3
L	L.4	1	1	4	2	5	6
M	M.1		1	3	1	4	4
M	M.2		1		1	1	1

C = Critical
Mj = Major
Mn = Minor



Divide To Priority Sets

❖ Order the priority using the defects density table

❖ Create Priority Sets:

- High
- Medium
- Low

Area	Test Group	Severity (Critical)	Severity (Major)	Severity (Minor)	Total (C+Mj)	Total (Mj+Mn)	Total
L	L.2	2	3	4	5	7	9
B	B.2	1	2	3	3	5	6
L	L.1	1	2		3	2	3
L	L.3	1	2		3	2	3
L	L.4	1	1	4	2	5	6
A	A.3		1	8	1	9	9
A	A.5		1	8	1	9	9
M	M.1		1	3	1	4	4
B	B.3		1	2	1	3	3
B	B.1		1	1	1	2	2
M	M.2		1		1	1	1
A	A.4			7	0	7	7
A	A.2			5	0	5	5
A	A.1			3	0	3	3



More Complex Analysis Table

- ❖ Analysis per tested **module or feature**
- ❖ Perform **deep analysis** per the various tests executed

Module/Feature	Risk Level	Defects (Sanity)	Defects (Dependencies)	Coverage Level in Previous Build	Infrastructure Impact	Average	Final Priority
Module A	4	2	2	1	2	2.2	Medium
Module B	3	4	4	4	4	3.8	High
Functionality A	4	5	3	3	1	3.2	High
Functionality B	1	1	1	3	1	1.4	Low



More Complex Analysis Table

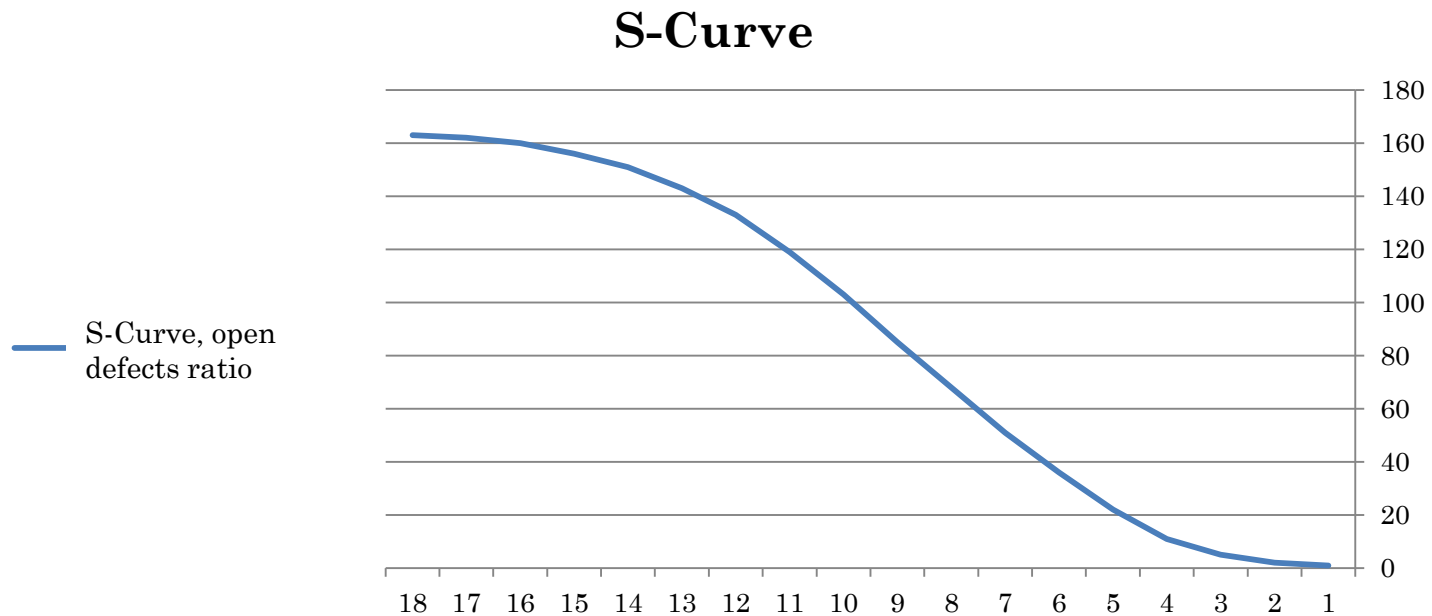
- ❖ Measure the infected area depending on **defects detection ratio**
- ❖ Prioritize the areas to **put the focus** when performing **Exploratory Testing**

Module/Feature	Risk Level	Defects (Sanity)	Defects (Dependencies)	Coverage Level in Previous Build	Infrastructure Impact	Average	Final Priority
Module A	4	2	2	1	2	2.2	Medium
Module B	3	4	4	4	4	3.8	High
Functionality A	4	5	3	3	1	3.2	High
Functionality B	1	1	1	3	1	1.4	Low



Defect Detection Ratio

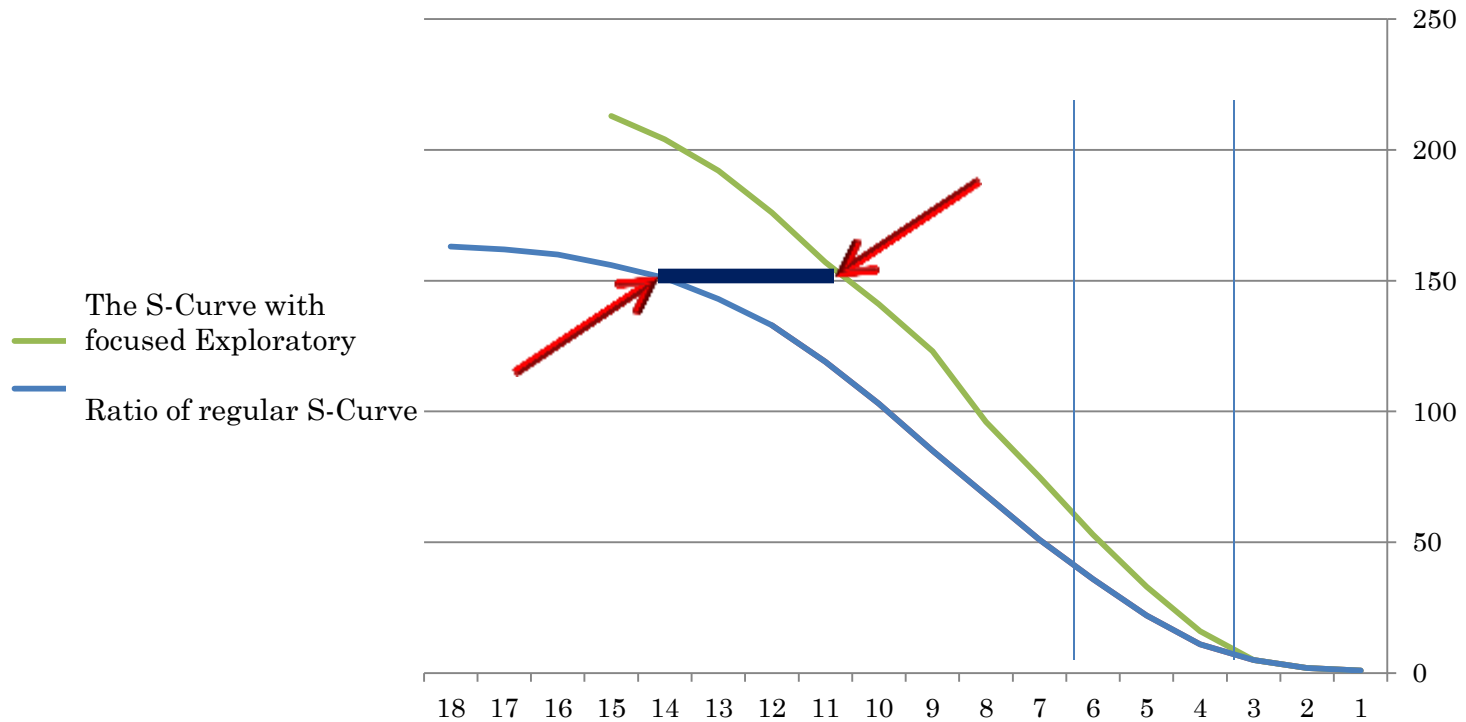
❖ Normal S-Curve





Improved Defect Detection Ratio

S-Curve using Focused Exploratory Testing





Where Do We See The Impact?



The Impact

- ❖ Approach method: **Defects driven** approach
- ❖ Project: **delivery timeline**
- ❖ Quality: **product's quality** raised
- ❖ Efficiency: **less time** invested at **defect detection**
- ❖ Positioning: **higher quality** of the Testing Team/s – **less escaped defects**



When To Expand The Focused Exploratory Tests?



When To Expand?

- ❖ At the time that we have **many features to test** and the **probability** to have **defects** is **high**
- ❖ At the time that we have **less time for detailed description** of the test cases
- ❖ At the time that we have **many change requests** that we **didn't cover with test cases**

- ❖ Recommendations:
 - Document the **skeleton** of the **Exploratory Tests** you executed
 - Describe in more details **those who encountered bugs** (-:



When To Reduce Or Not Perform The Focused Exploratory Tests?



When To Reduce Or Not Perform?

- ❖ At the time that we have **only minor areas to cover**,
- ❖ At the time that **we see defect convergence** – less bugs open,
- ❖ At the time that we see that there are **less areas that are getting “infected”**,
- ❖ At the time that the **ROI is less effective**.

Agenda

Introduction

Presentation objectives

The challenges that we have

Take control over the process

Summary





Before The Summary... What are the Next Steps?



Next Steps

- ❖ Try to Implement that in an **Agile** environment,
- ❖ Start to use **Heuristic** methods!
- ❖ Where do we find most of our **bugs**? (ODC analysis)
- ❖ How to reduce **over-testing**?
- ❖ Etc' (-;



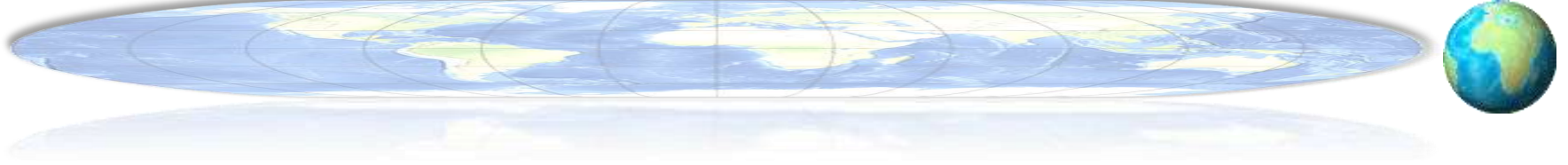
Summary

- ❖ **Analyze your testing project** and decide if to **activate this approach**
- ❖ Feel free to **Mix** some **Execution Approaches**
- ❖ Factors for **successful implementation**:
 - **Initial risk analysis**
 - **Prioritize** test areas – High, Medium and Low
 - **Daily defect detection analysis** – per area
 - Focus on areas with more **potential** to have **defects**
 - **Communicate** plan and results with your **peers** and **managers**
- ❖ **Good Luck!**



THANK YOU





Q & A

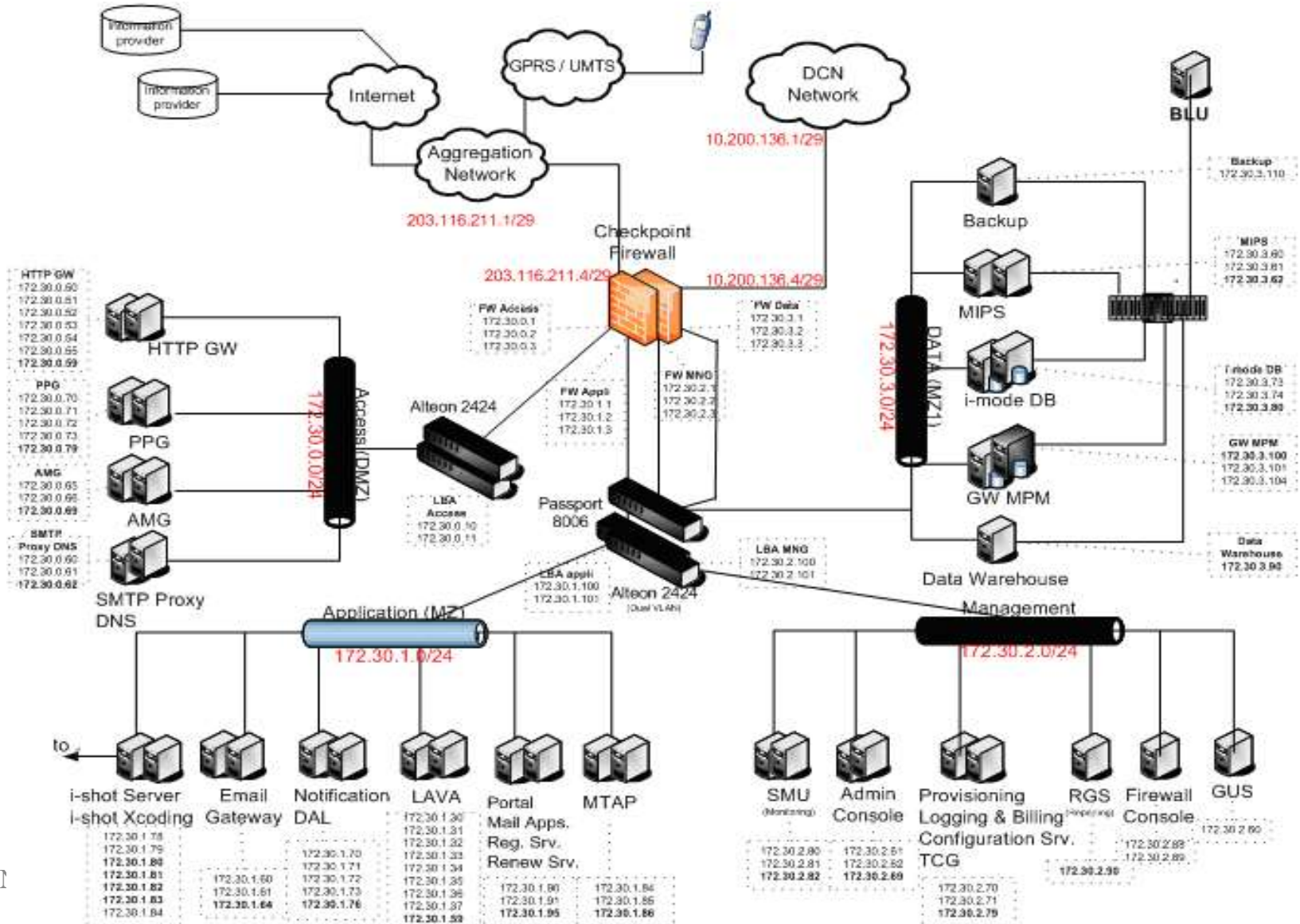
Yaron Tsubery:

yaronts@smartest.com

Office: +972 2 6509255

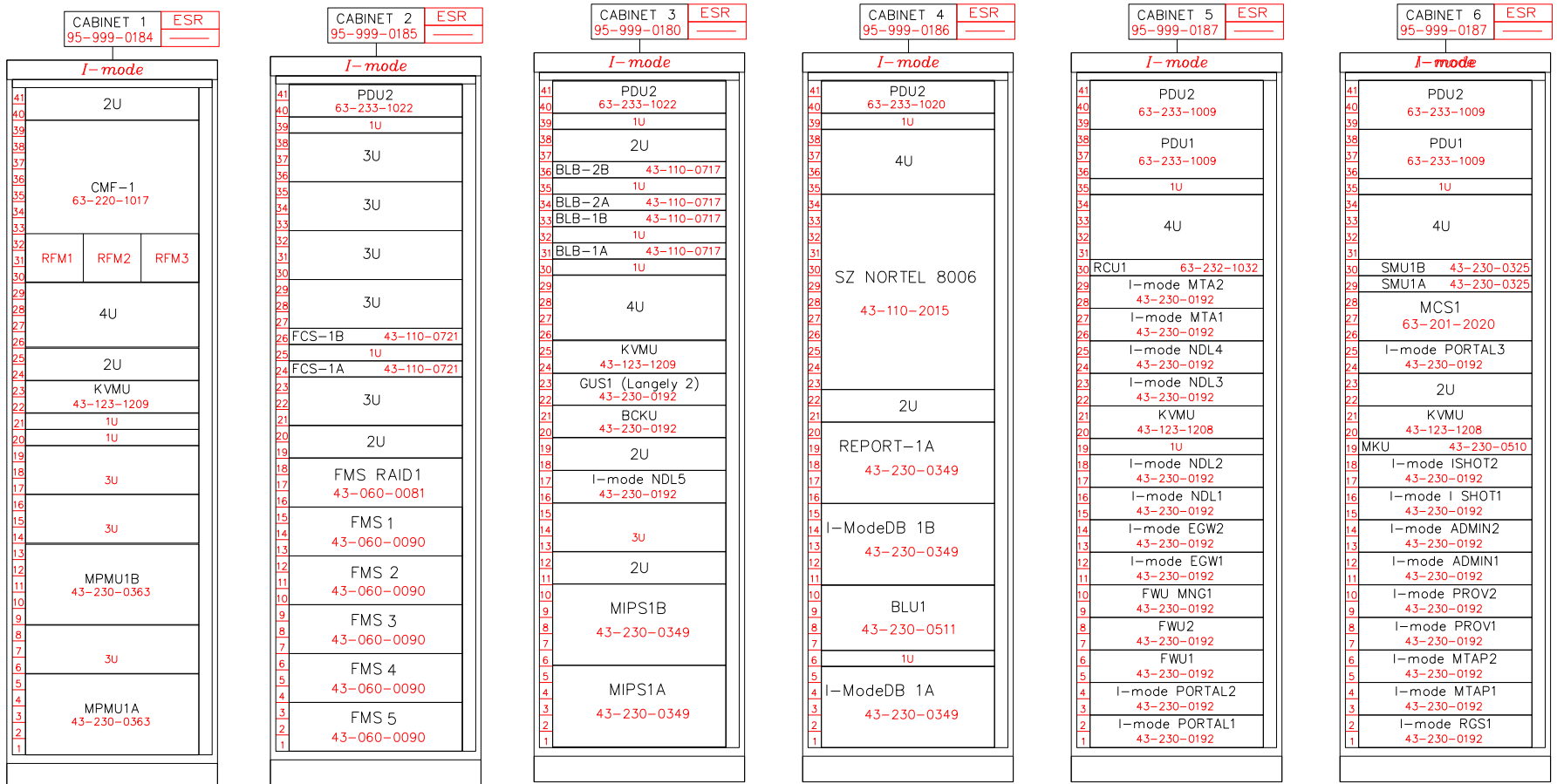


System: Physical Architecture



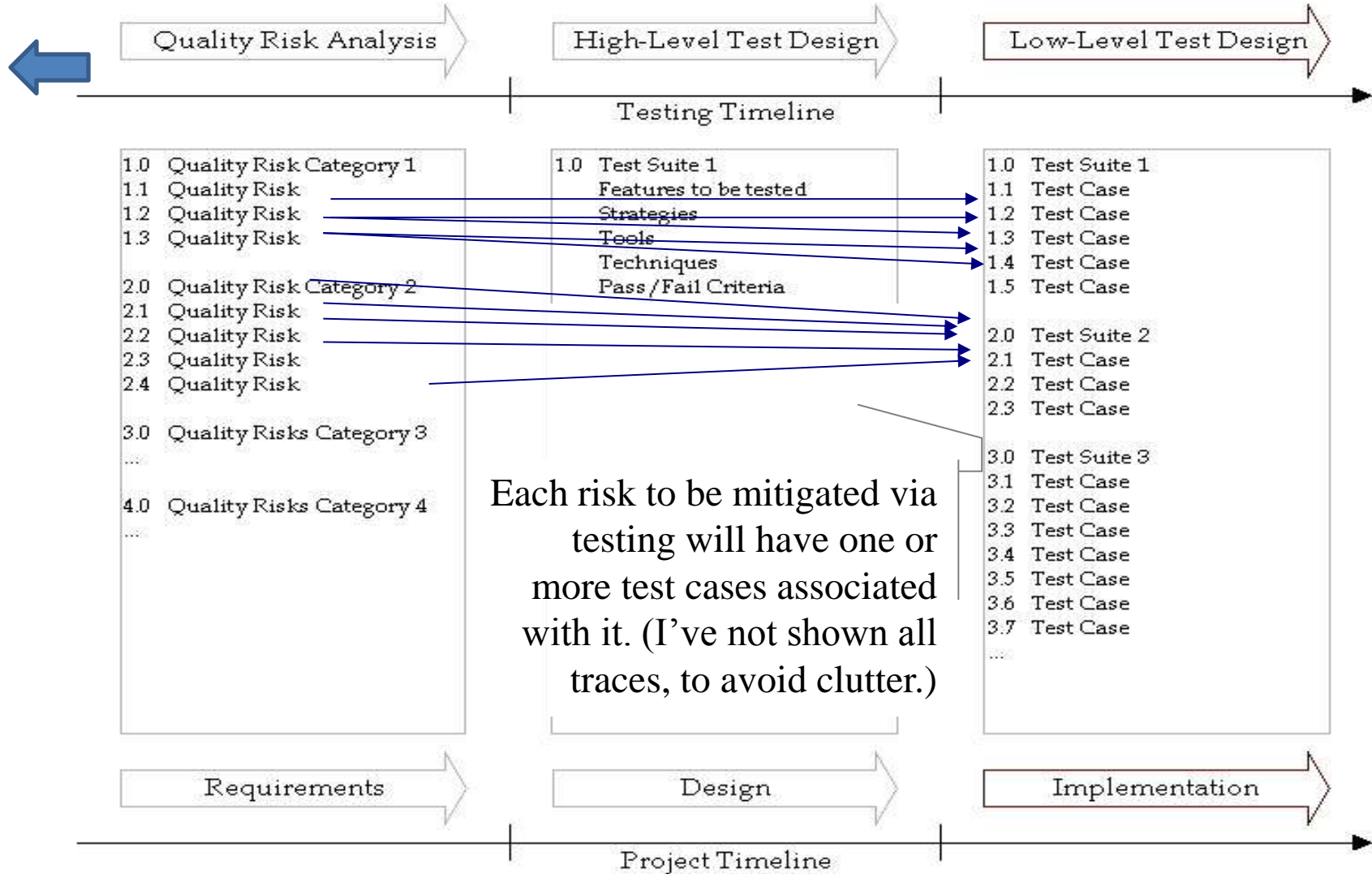


System Layout





Low-Level Design





Quality risks are potential system problems which could reduce user satisfaction

Risk priority number: Aggregate measure of problem risk

Business (operational) risk: Impact of the problem

Technical risk: Likelihood of the problem

Tracing information back to requirements, design, or other risk bases

Quality Risk

Risk Category 1

Risk 1

Risk 2

Risk *n*

Tech.
Risk

Bus.
Risk

Risk
Pri. #

Extent of
Testing

Tracing

A hierarchy of risk categories can help organize the list and jog your memory.

- 1 = Very high
- 2 = High
- 3 = Medium
- 4 = Low
- 5 = Very low

The product of technical and business risk, from 1-25.

- 1-5 = Extensive
- 6-10 = Broad
- 11-15 = Cursory
- 16-20 = Opportunity
- 21-25 = Report bugs