

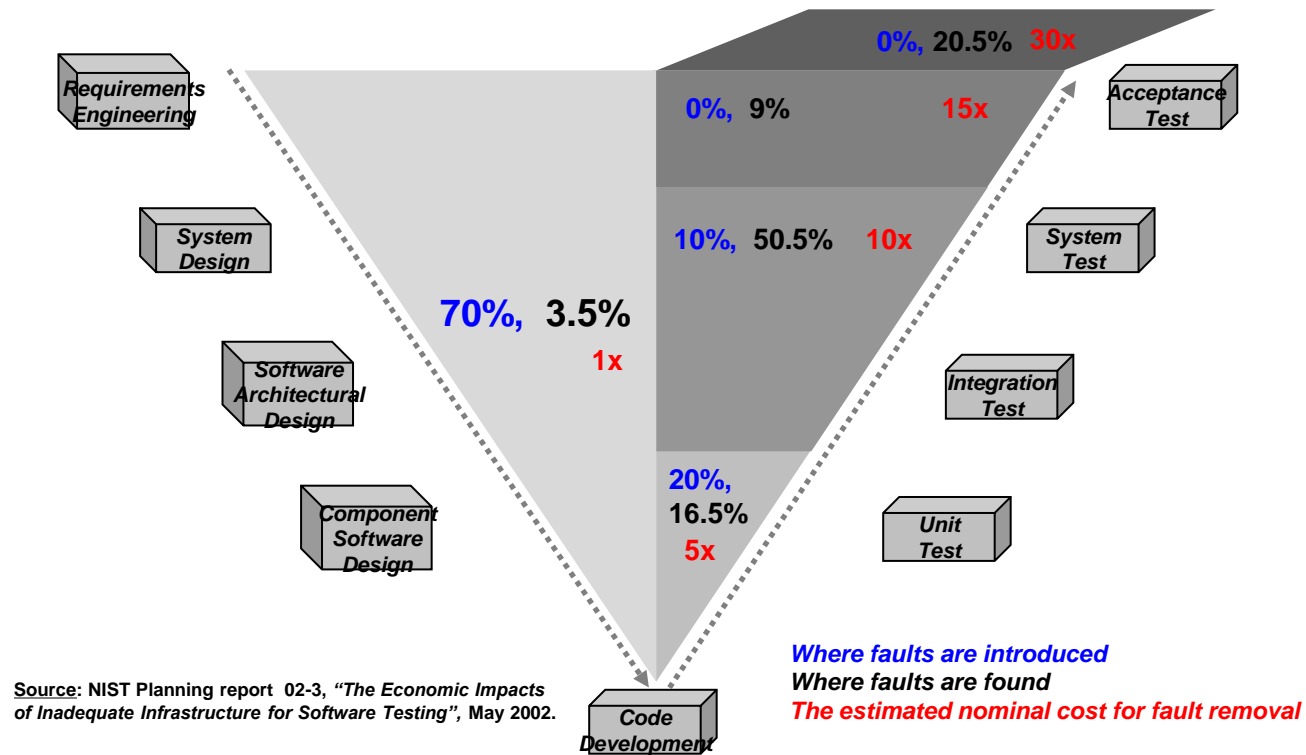
# O-DA version up working group

An introduction of a project for version up of  
Open Group Standard -

Dependability through Assuredness™ (O-DA) Framework

# Background of new version of O-DA

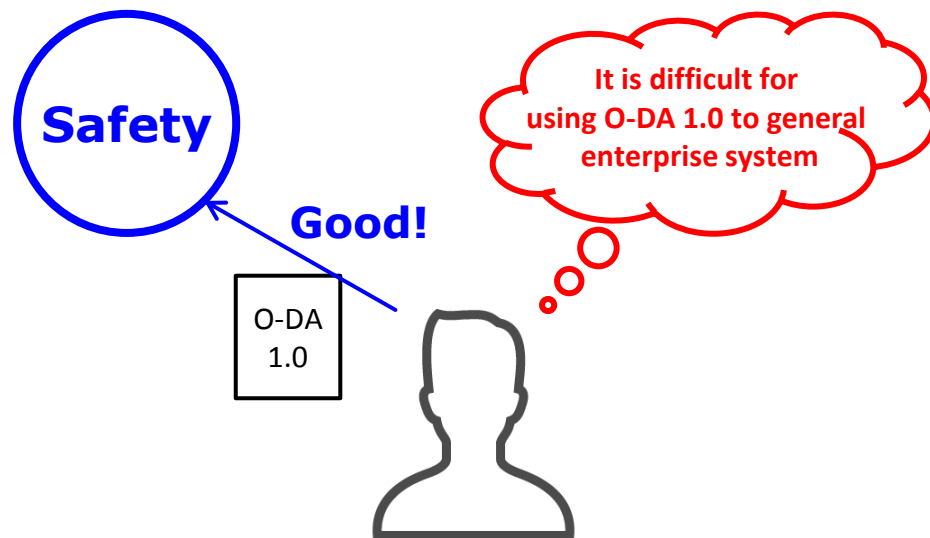
Although many defects are introduced into a system before testing phases, most of these defects can not be detected at these phases.



# Background of new version of O-DA




Hence, we aim to extend O-DA 1.0 to improve the dependability of general enterprise system which includes early detecting the defects of the system.

O-DA 1.0 has been used for “Safety” mainly



Agile is one of solutions against the issue

But, Agile can not always assure a dependability of general enterprise systems

-  Some of stakeholders aren't interested in Non Functional Requirements (NFRs)
-  Some of stakeholders are lacked
-  Jr. developer can't imagine irregular cases

It depends on stakeholder's knowledge and experience

# Master Schedule












No.	Action Item	'18	2019												2020													
		12	01	02	03	04	05	06	07	08	09	10	11	12	01	02	03	04	05	06	07	08	09	10	11	12		
1	Planning	★																										
2	Learning O-DA1.0 (new comer)		▶																									
3	9 GRIDS	Glossary & Sentence Structure																										
4		Business Value	▶																									
5		Business Functions	▶																									
6		Migration and Operation																										
7		Performance																										
8		Security																										
9		Test Management																										
10		Agile/DevOps	▶																									
11		Open Data																										
12	Gathering The Evidence		▶																									
13	Documentation		▶																									
14	International Conference		★				★						★					★								★ Launch		
15	Discussion of O-DA2.0 Use Cases		▶																									
16	Adoption of Archimate to O-DA2.0		▶																									

# Master Schedule (Action Items)

No.	Action Item	Explanation	
1	Planning	To plan about The project of O-DA version up	
2	Learning O-DA1.0 (new comer)	To learn about O-DA1.0. If a member don't know about what are written in O-DA1.0, the member can't understand that what does he/she should write into O-DA2.0. Therefore member who don't know O-DA1.0 should learn about 1.0. A member who understand O-DA1.0 don't need learn it.	
3	9 GRIDS	Glossary & Sentence Structure	To develop a Verifying Techniques of a Glossary and Sentence Structure of documents of a target system
4		Business Value	To develop a Business Value Analysis Techniques and Viewpoints from a customer pain point perspective: Before and After developing a target system.
5		Business Functions	To develop a Viewpoints for detecting the defects in Business Functions in earlier phases or timing.
6		Migration and Operation	To develop a Viewpoints for detecting the defects about Migration and Operation in earlier phases or timing.
7		Performance	To develop a Viewpoints for preventing or mitigating a Performance issues before it will be occurred.
8		Security	To develop a Viewpoints for preventing or mitigating a Security issues before it will be occurred.
9		Test Management	To develop a Techniques of Test management to assure dependability while improving test efficiency.
10		Agile/DevOps	To develop a Adoption techniques of O-DA2.0 to Agile projects and DevOps.
11		Open Data	To develop a Data Format for measure an effectiveness of O-DA2.0. It will be able to compare to other data.
12	Gathering The Evidence	To gather evidences for demonstrate of O-DA2.0 effectiveness.	
13	Documentation	To write a standard document (includes White paper) of O-DA2.0.	
14	International Conference	To attend the International conferences of The Open Group for discuss about O-DA2.0. Finally we will launch the O-DA2.0.	
15	Discussion of O-DA2.0 Use Cases	To discuss about how to use the O-DA2.0 from a lot of perspective of User company, IT Vendors and Consulting firms etc.	
16	Adoption of Archimate to O-DA2.0	To define and testing how to represent O-DA2.0 by Archimate.	

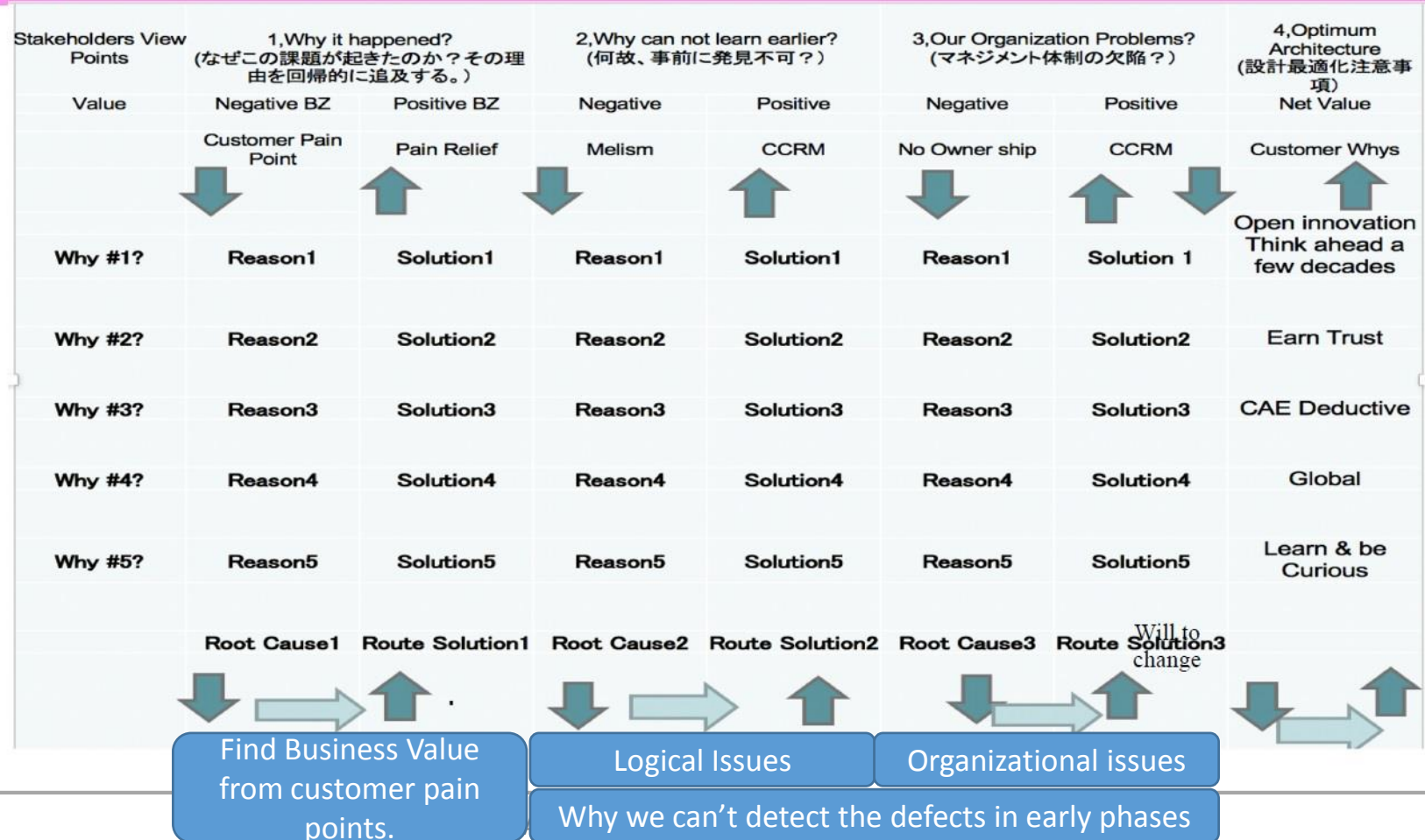
# Contents of 9GRIDS

We will develop below the 9 GRIDS for detect earlier the defects of system/software.

No.	Category	GRIDS	Explanation	Deliverables
1	Common	<b>Glossary and Sentence Structure</b>	A Verifying Techniques of a Glossary and Sentence Structure of documents of a target system (requirement / functional design / etc. ), such as 5W1H.	Guide Format  
2	Purpose	<b>Business Value</b>	Business Value Analysis Techniques and Viewpoints from a customer pain point perspective: Before and After developing a target system.	Guide Viewpoint  
3	Functional Requirements	<b>Business Functions</b>	A Viewpoints for detecting the defects in Business Functions in earlier phases or timing.	Viewpoint 
4		<b>Migration and Operation</b>	A Viewpoints for detecting the defects about Migration and Operation in earlier phases or timing.	Viewpoint 
5	Non Functional Requirements	<b>Performance</b>	A Viewpoints for preventing or mitigating a Performance issues before it will be occurred.	Viewpoint 
6		<b>Security</b>	A Viewpoints for preventing or mitigating a Security issues before it will be occurred.	Viewpoint 
7	Methodology	<b>Test Management</b>	A Techniques of Test management to assure dependability while improving test efficiency.	Guide 
8		<b>Agile/DevOps</b>	Adoption techniques of O-DA2.0 to Agile projects and DevOps.	Guide 
9	Measure	<b>Open Data</b>	A Data Format for measure an effectiveness of O-DA2.0. It will be able to compare to other data.	Format 

# Appendix. 5 Whys Backward reasoning

## O-DA 5 Whys for Customer Pain points Back ward reasoning (JFujieda 2018 CCRM EA Backward Reasoning Template)



# Appendix. Test Management

To consider that how to manage which can be assured the dependability of the system by manage the Building Blocks (BBs) which are verified and assured and the combination so that the test volume does not explode even if the number of modules increases.

