Software and Computing CoDR

2.5 Risks and Management of Risks

2012 Feb 16

D. Hall, SPDO

Introduction

Assessed risk exposures

Comparison with SEI risk taxonomy

High level analysis

Plans to manage risks and issues

Risk exposure thresholds define exposure levels for CoDR

Likelihood:

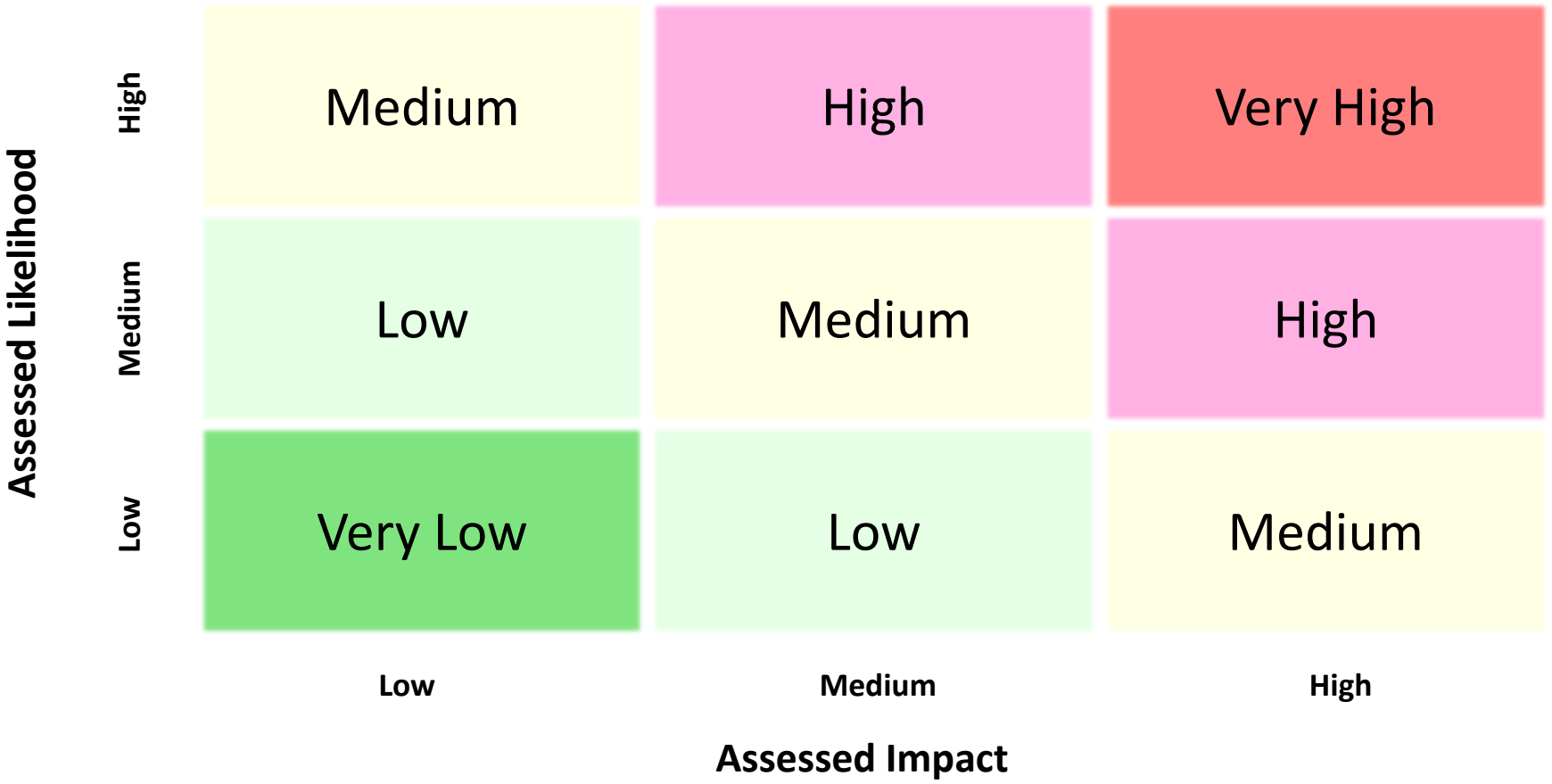
- Low : less than estimated 30% likelihood of occurrence, [7] Risk Categories 0-2
- Medium : at least 30% but less than 50% likelihood of occurrence, [7] Risk Categories 3-5
- High : at least 50% likelihood of occurrence, [7] Risk Categories 6-10

Impact:

Impact	Cost	Schedule	Performance
Low	Less than 10% impact, [1] Impact levels 1-3	Very minor or no slip in milestone, i.e. order one month	Very minor or no impact
Medium	Order 10% to 20% impact, [1] Impact level 4	Moderate slip in milestone, i.e. up to 6 months	Moderate functional impact or reduction in performance, performance almost acceptable but would require redesign
High	Greater than 20% impact, [1] Impact level 5	Critical slip in milestone, i.e. more than 6 months	Critical functional impact or reduction in performance, performance not acceptable and requires new design

Table 1 Risk Impact Definitions

Resulting risk exposure assessments: from ‘very low’ to ‘very high’



Risks Related to Software Engineering
Note: levelling across all risks yet to be done

No.	Risks	Short Description	Risks Becoming Issues Results in:	Proposed Plans to Manage Risks & Issues *Current Status*	Risk Owner	Impact: SKA1 *SKA2*	Likelihood: SKA1 *SKA2*	Exposure: SKA1 *SKA2*
1.1	Disassociation between Monitoring and Control and other software-intensive elements of the system	• Insufficient interface definition between – and integration with – other software implementations	• Unnecessary re-work • Under-estimates of cost, time and the resources required to meet the requirements of software and computing • Severe and negative impact on the project as a whole	• Establish and maintain mechanisms to capture and assess early signs of negative scope risk *Current Status:* • *Risks documented* • *Issues not yet manifest*	TPM, WPC	High *High* **SKA1**	High *High*	Very High *Very High*
1.2	A wide variety of "antipattern" behaviours	[5] "In software engineering, an antipattern is a pattern that may be commonly used but is ineffective and/or counterproductive in practice."	Under-estimates of cost, time and the resources required to meet the requirements of software and computing. Negative impact on the project as a whole due to inappropriate requirements analysis.	• Adherence to the [1] SEMP • Continual monitoring and management of potentially dysfunctional behaviours • Apply internationally recognised standards and good practices for software development appropriate to the SKA software development effort • Learn from Precursor and Pathfinder experiences *Current Status:* • *Risks documented* • *Issues not yet manifest*	TPM, WPC	Medium *High*	Medium *Medium*	Medium *High*
1.3	Misinterpretation and erroneous analysis of requirements	The flow down of requirements is open to misinterpretation particularly when this is solely via document handover	Delivered designs may not meet the original intention of the requirement	• Use agreed common processes and tools to share information related to requirements • Close collaboration between the parties involved in generating requirements including regular reviews of requirements • Learn from Precursor and Pathfinder experiences *Current Status:*	TPM, WPC	High *High* ***SKA2***	Medium *Medium*	High *High*

Introduction

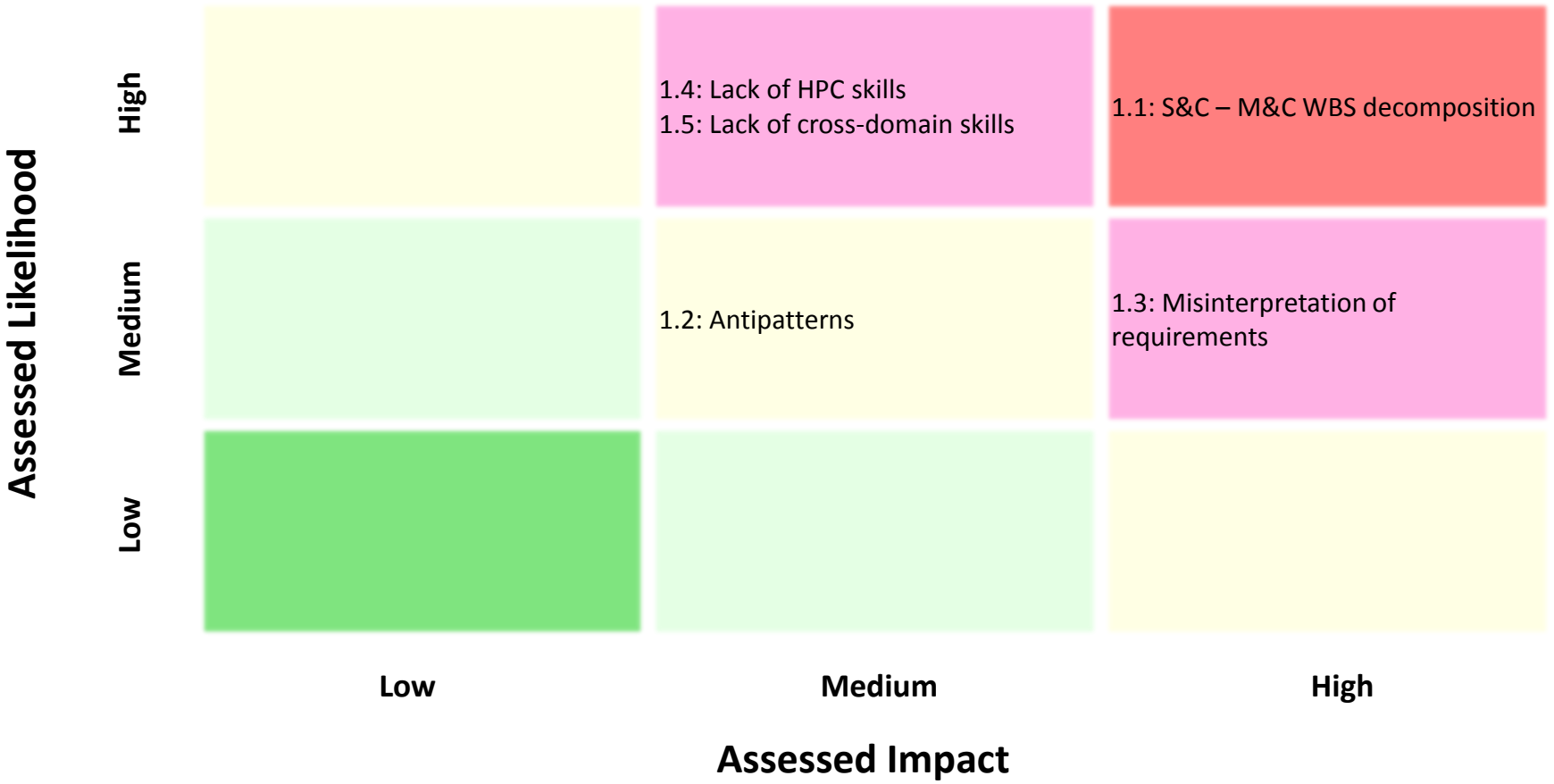
Assessed risk exposures

Comparison with SEI risk taxonomy

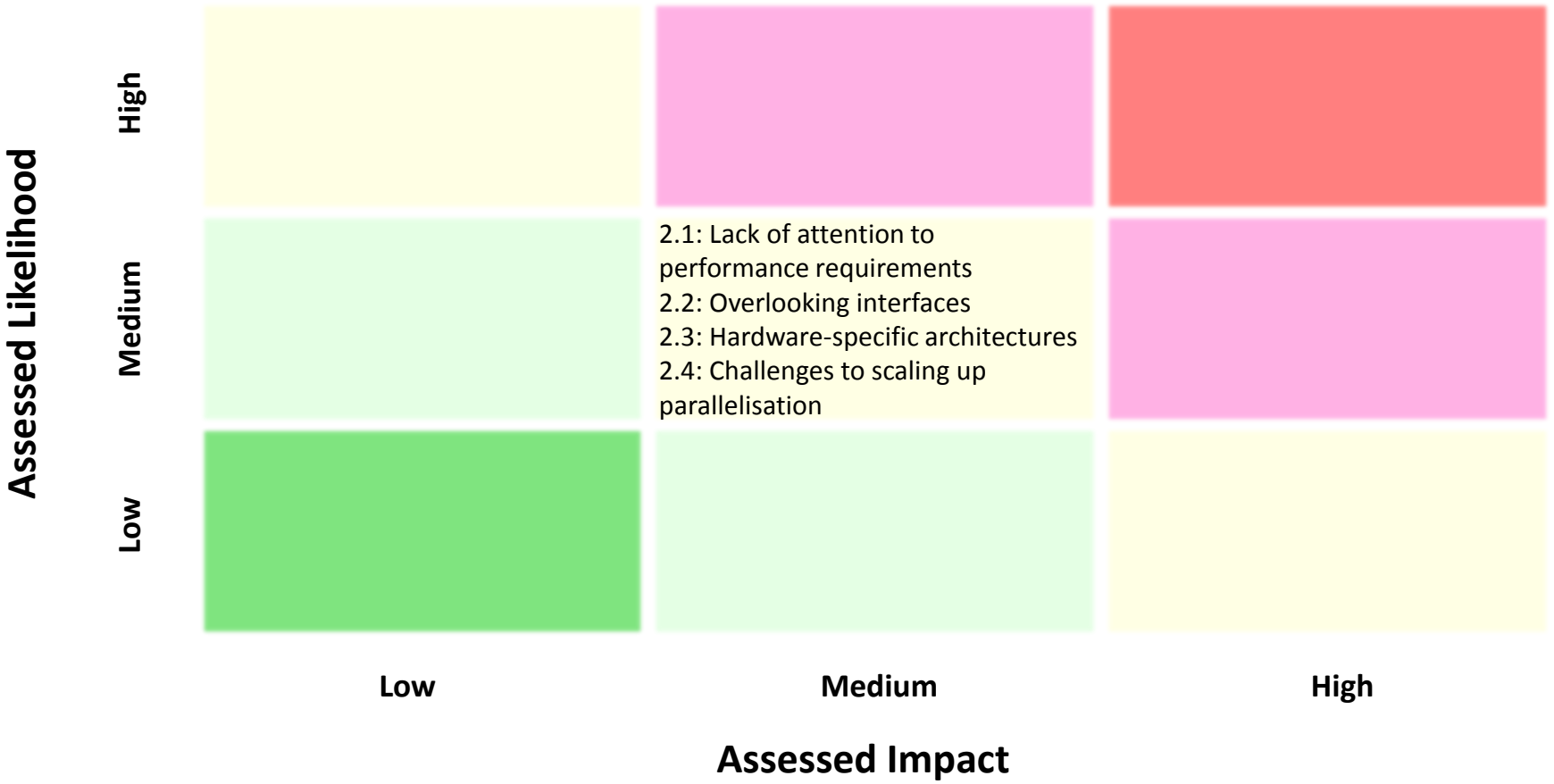
High level analysis

Plans to manage risks and issues

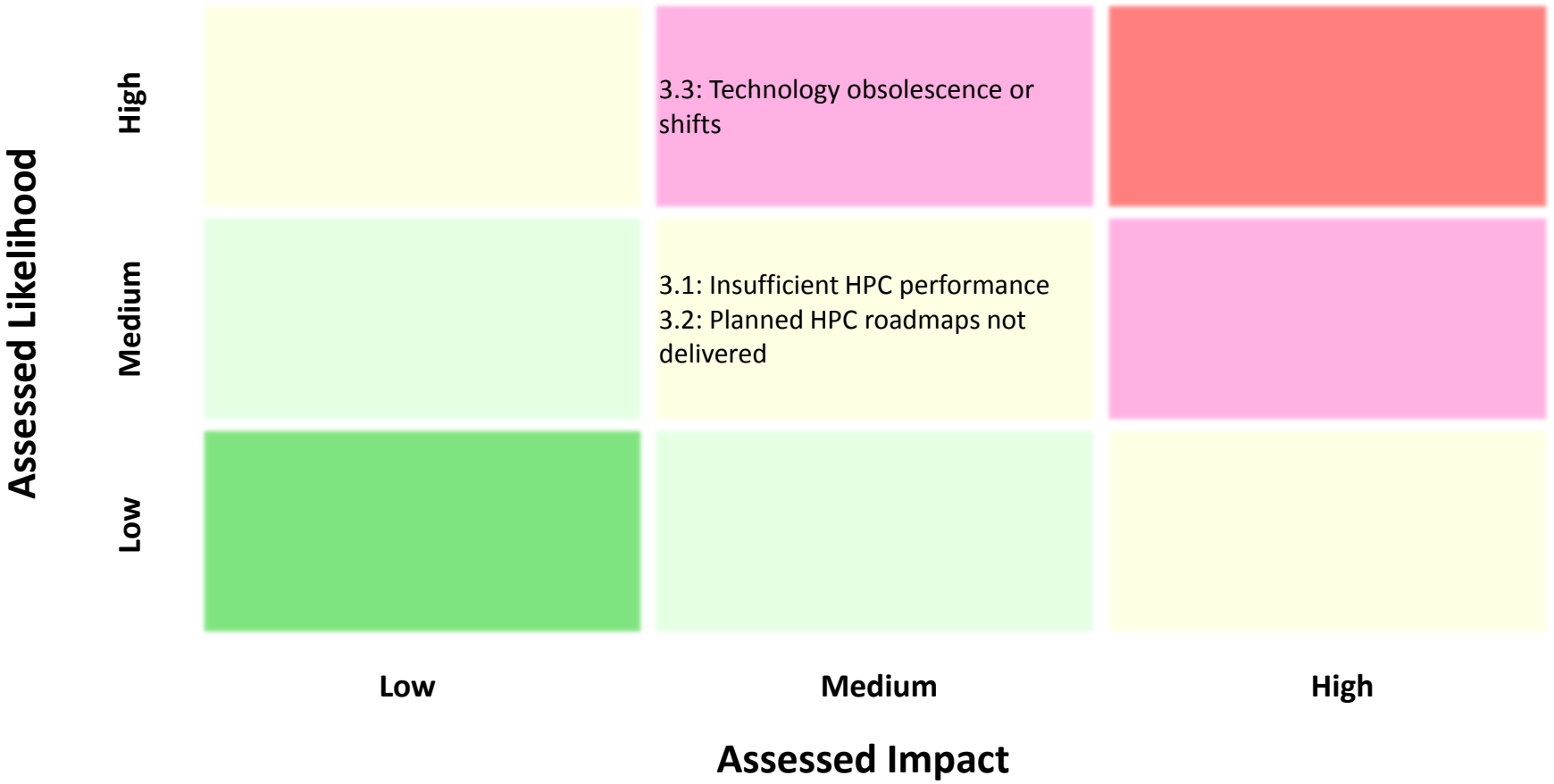
1: SKA_1 risks related to Software Engineering



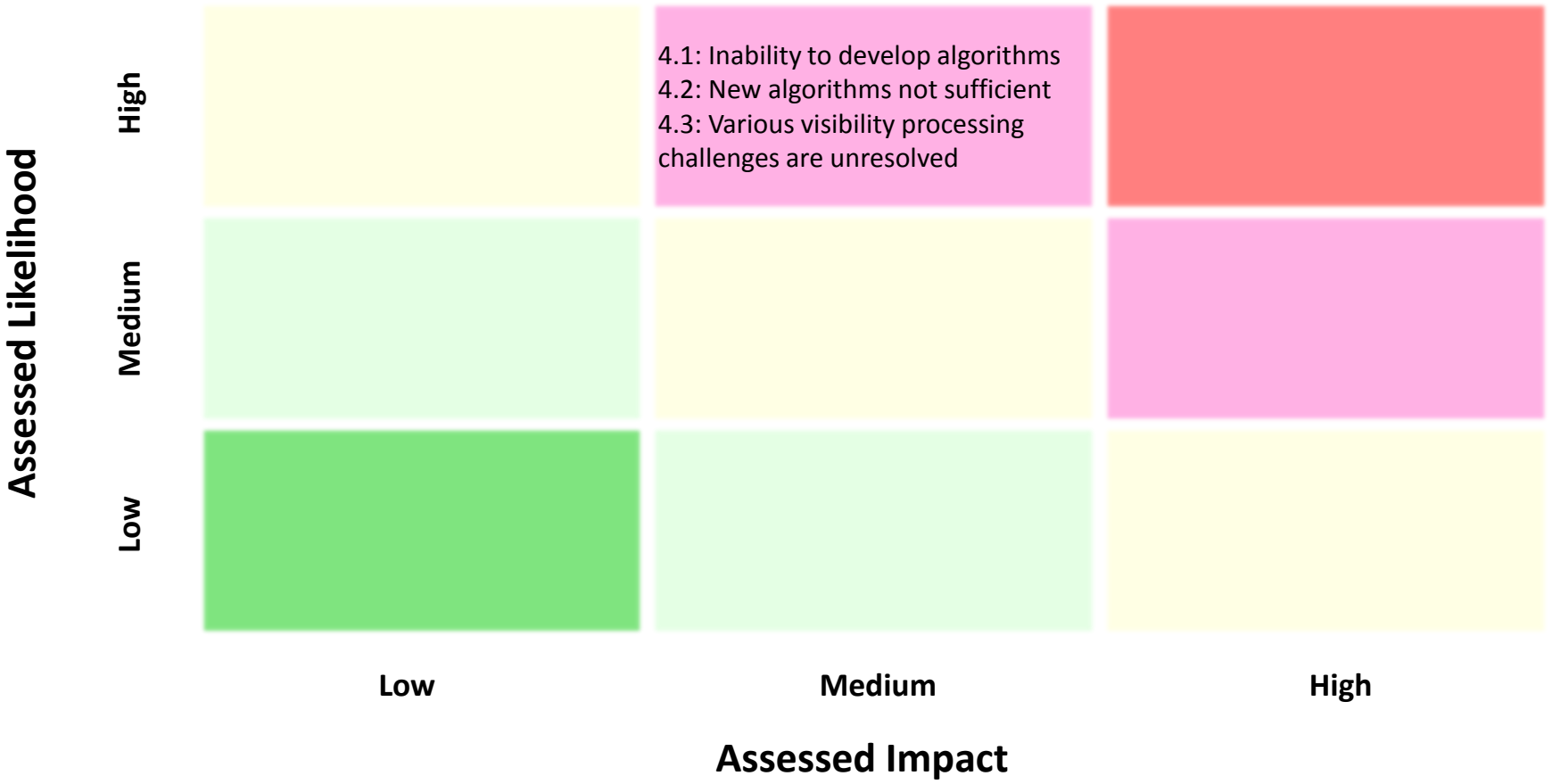
2: SKA_1 risks related to Software Architecture



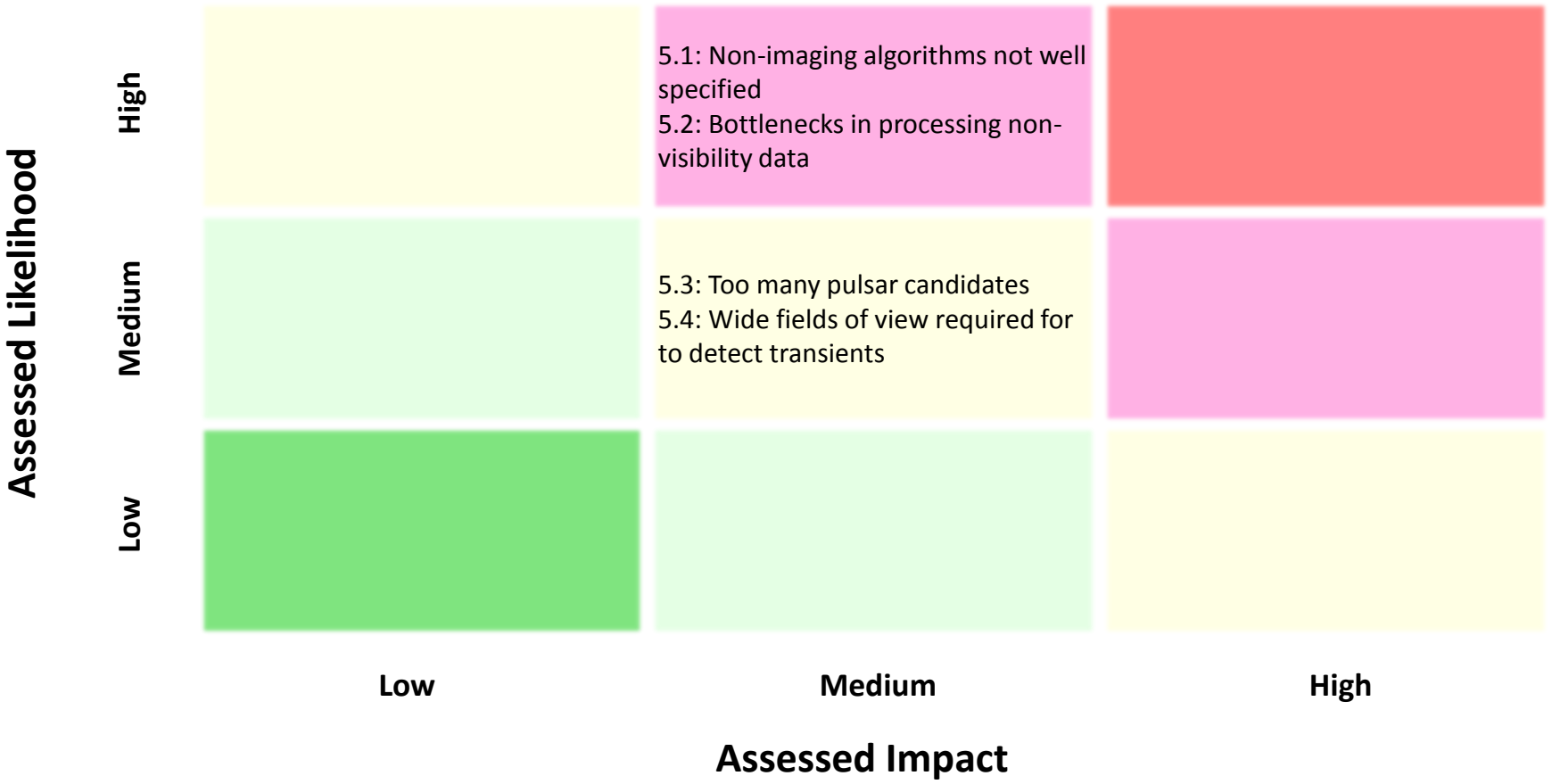
3: SKA_1 risks related to HPC Hardware



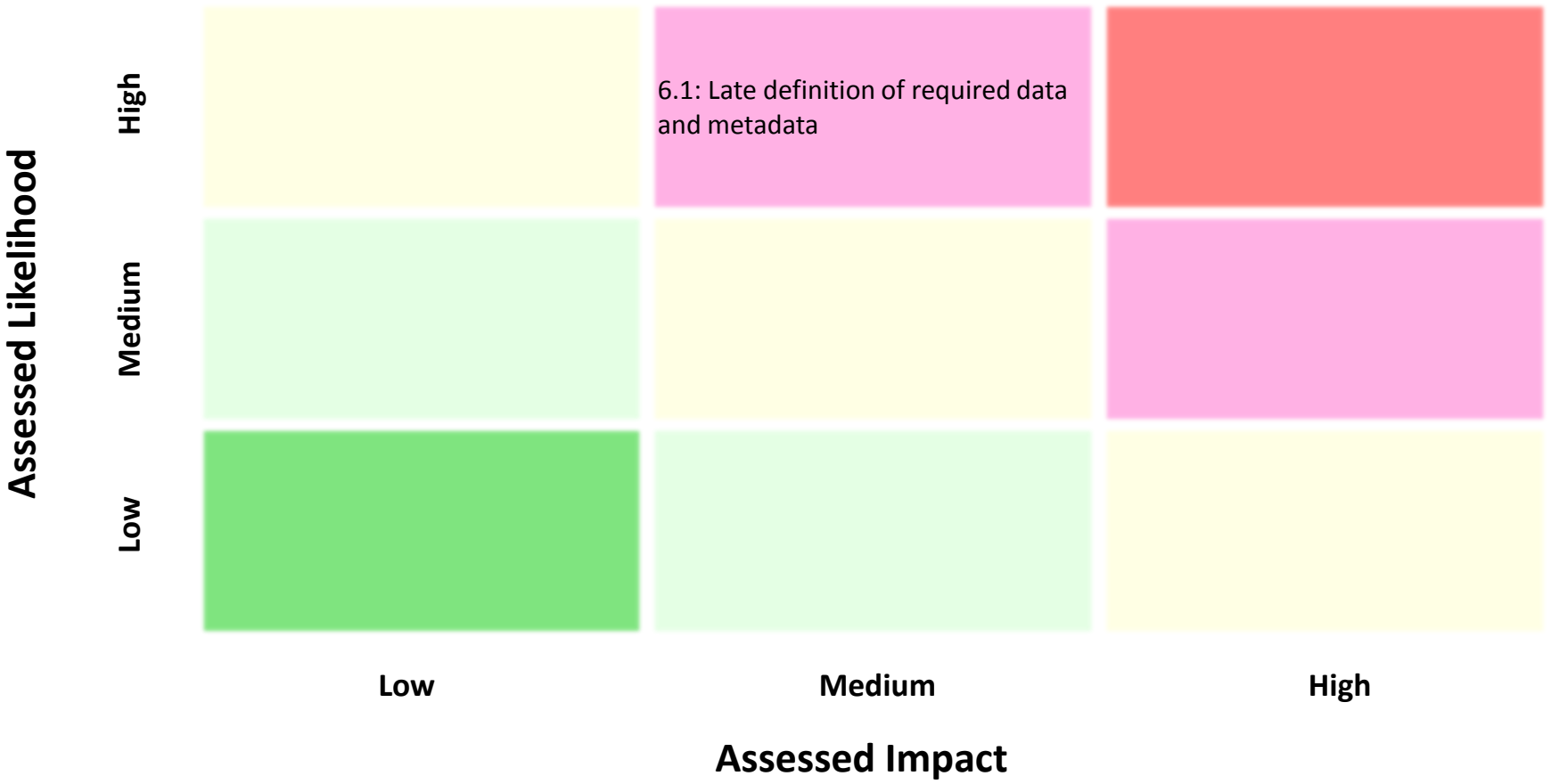
4: Unable to achieve visibility processing goals for SKA_1



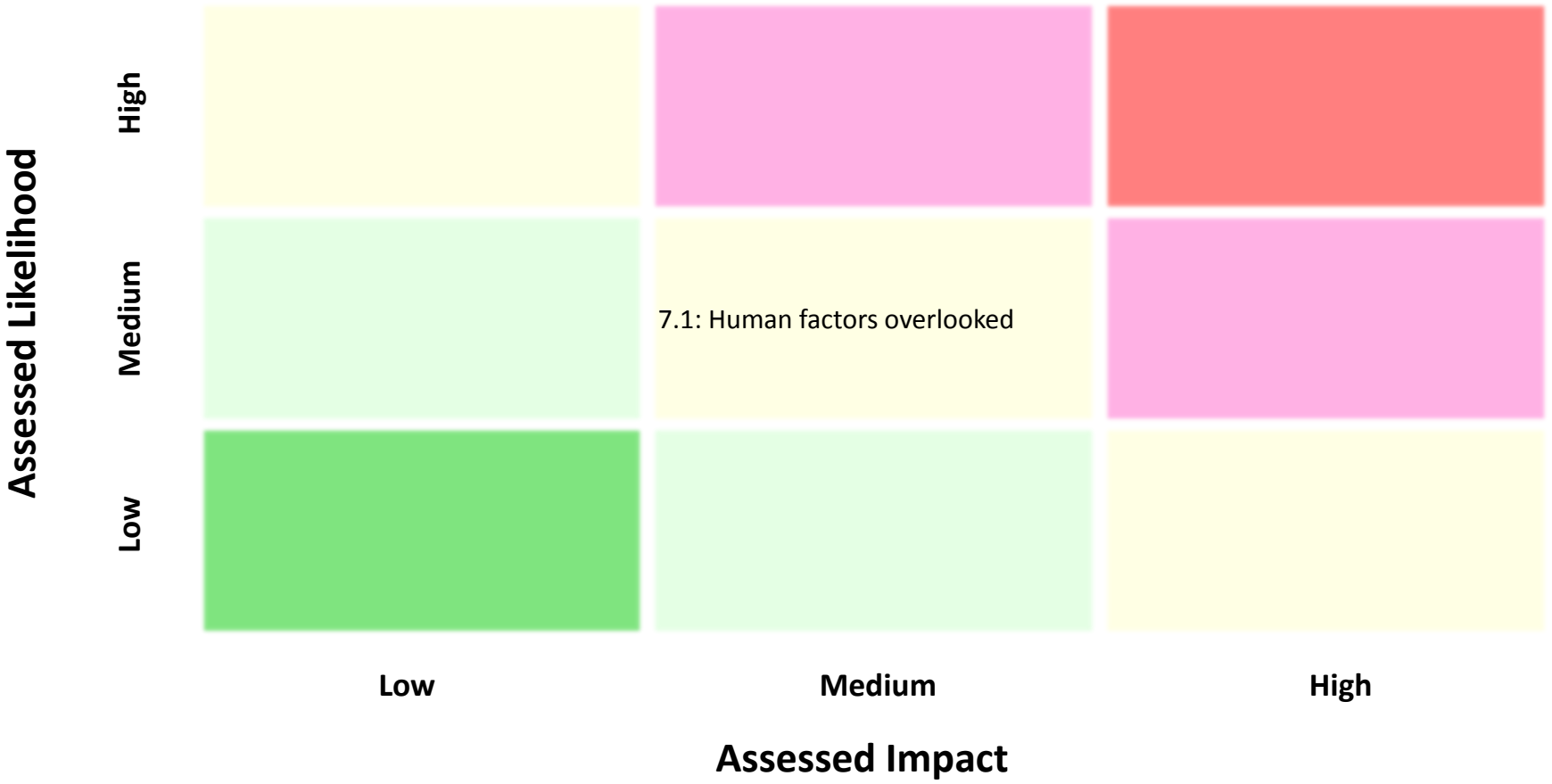
5: Unable to achieve non-visibility processing goals for SKA_1



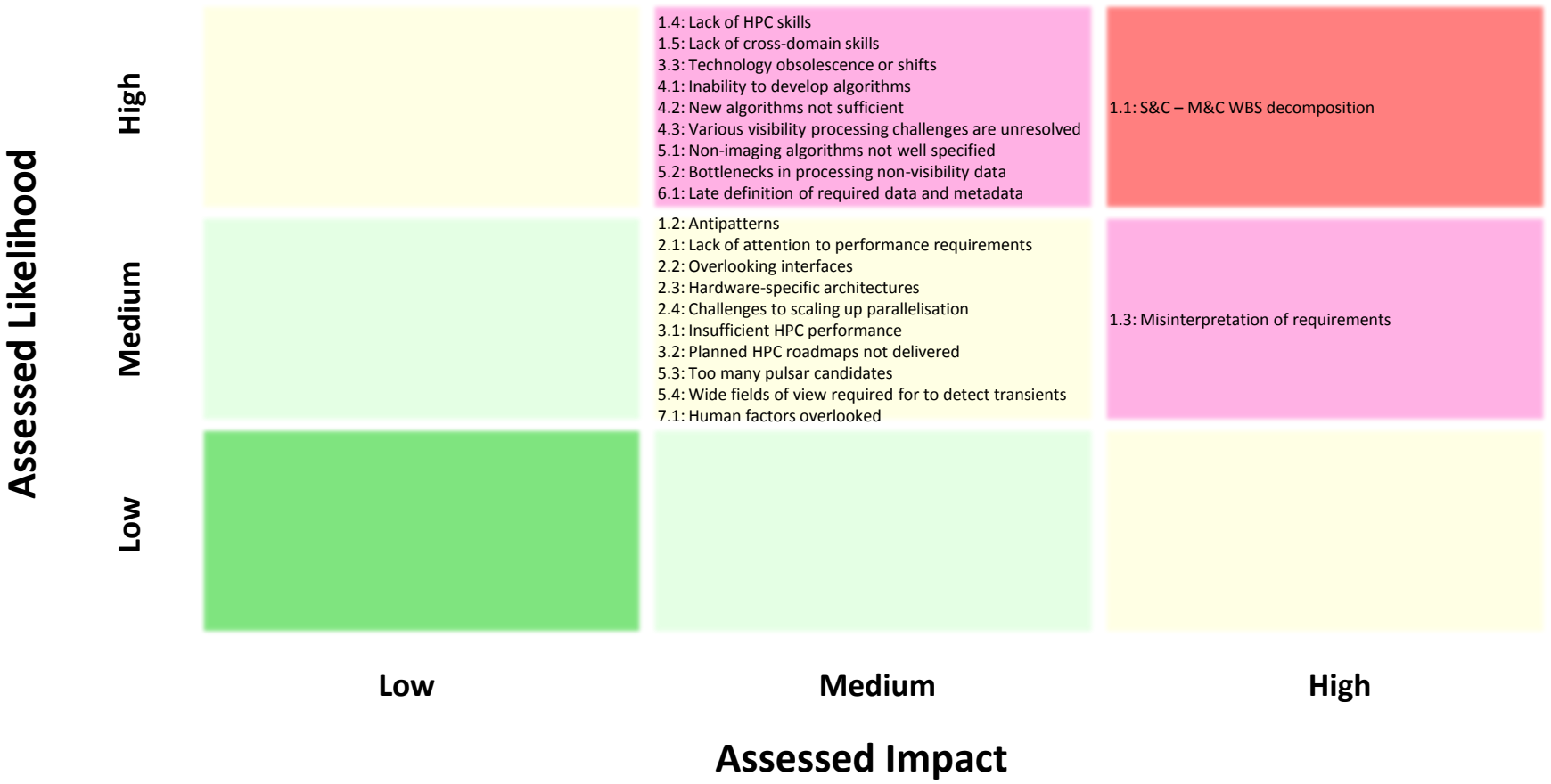
6: SKA_1 risks related to data products, storage and distribution



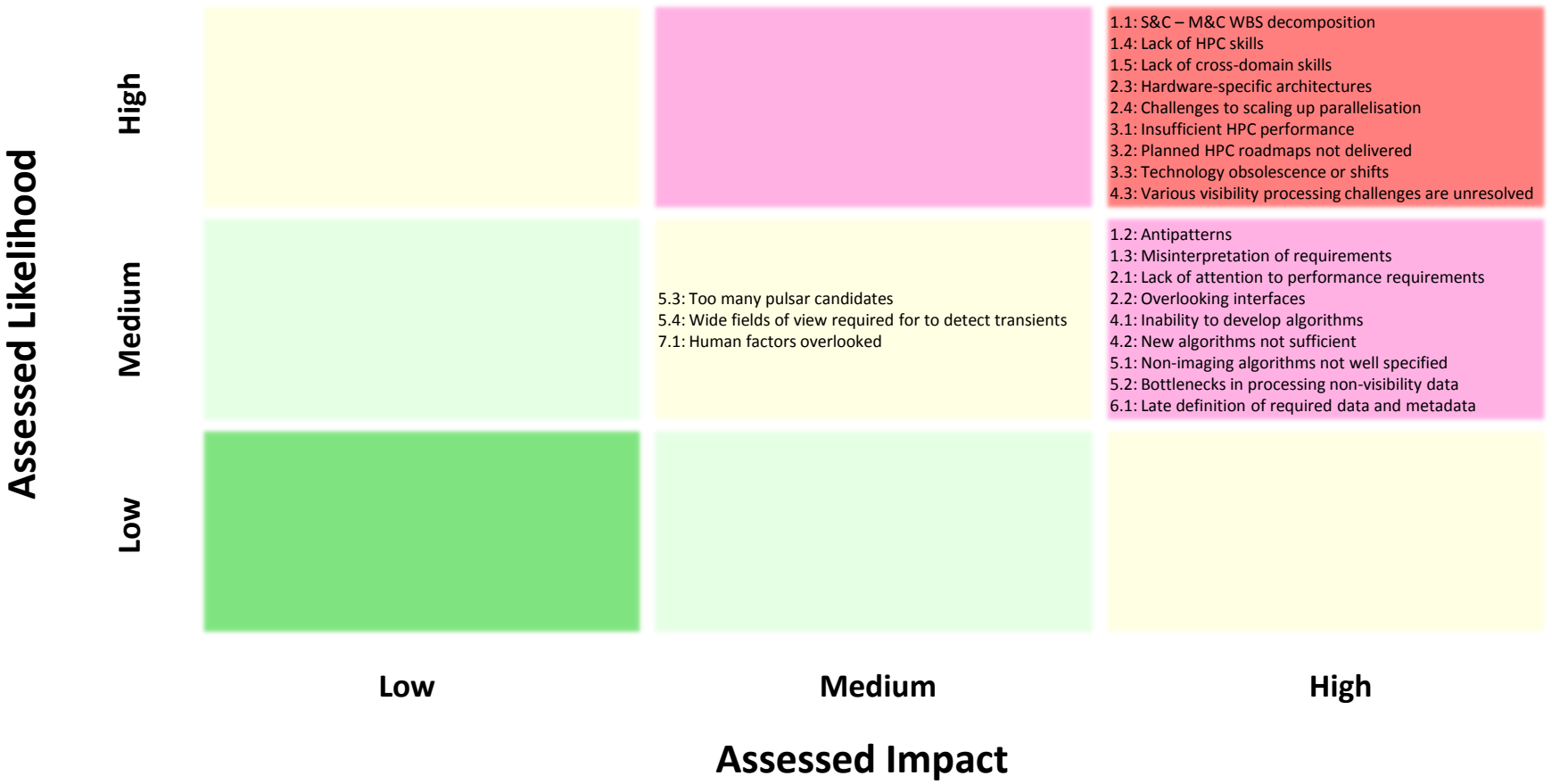
7: SKA_1 risks related to interfaces for users and operators



Summary of SKA_1 risks presented in Software and Computing CoDR Risk Register



Assessed SKA_2 overall risk impact & exposure is greater than for SKA_1, primarily due to scale



Introduction

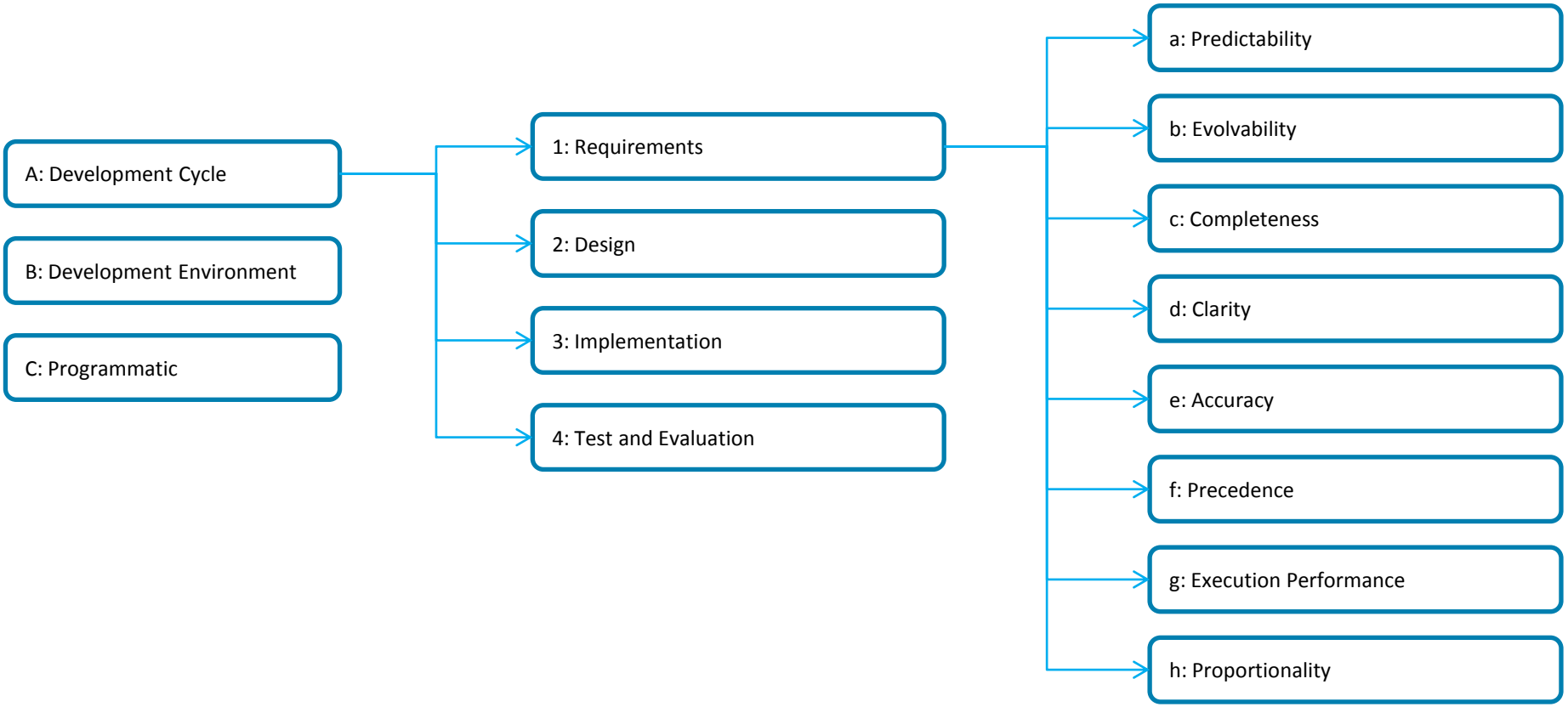
Assessed risk exposures

Comparison with SEI risk taxonomy

High level analysis

Plans to manage risks and issues

SEI taxonomy of risks for HPC application development – example decomposition



S&C Risk Register – links to SEI taxonomy of HPC development cycle risks

WP2-050.020.010-RE-001 / CMU-SEI_2006-TN-039					1.0 Software Engineering					2.0 Software Architecture				3.0 Computing and Exascale			4.0 Unable to Achieve Science Goals for Visibility Data Processing			5.0 Unable to Achieve Science Goals for Non Visibility Data Processing				6.0 Data Products, Data Storage and Data Distribution	7.0 Interfaces for Users and Operators
					1.1 Disassociation between Monitoring and Control and other software-intensive elements of the system	1.2 A wide variety of "antipattern" behaviours	1.3 Misinterpretation and erroneous analysis of requirements	1.4 Limited availability of software engineering resources for exascale	1.5 Limited availability of cross-domain experts	2.1 Lack of attention to non-functional performance requirements	2.2 Overlooking interfaces	2.3 Hardware- specific software architectures	2.4 Challenges in continuing to scale up the parallelisation of codes by many orders of magnitude	3.1 Scaling: sufficiently powerful computers in appropriate price range to deliver science results are not available when needed	3.2 MPublished HPC development roadmaps are not delivered	3.3 Obsolescence and technology paradigm shifts	4.1 Algorithm and implementation development; coupled with limited availability of developers	4.2 An enhanced path of continued algorithm development does not deliver performance expectations sufficient to deliver SKA2	4.3 Various imaging-related considerations not yet demonstrated as having been solved	5.1 Implementations are not well specified with large potential impact on processing systems.	5.2 Non visibility data processing	5.3 Too high a level of candidate pulsar detections	5.4 Wide fields of view are required to detect transients	6.1 Definitions of required data products and associated metadata products are late	7.1 Human factors are overlooked in developing interfaces
A	Development Cycle	1	Requirements	a	Predictability																				
				b	Evolvability																				
				c	Completeness																				
				d	Clarity																				
				e	Accuracy																				
				f	Precedence																				
				g	Execution Performance																				
				h	Proportionality																				
		2	Design	a	Difficulty																				
				b	Modularity																				
				c	Usability																				
				d	Maintainability																				
				e	Portability																				
				f	Reliability																				
		3	Implementation	a	Specifications																				
				b	Project Plan																				
				c	Scale of Effort																				
		4	Test and Evaluation	a	Verification: (i) Unit Testing																				
				b	Verification: (ii) Integration																				
				c	Verification: (iii) Interoperability																				
				d	Validation																				

S&C Risk Register – links to SEI taxonomy of HPC development environment risks

WP2-050.020.010-RE-001 / CMU-SEI_2006-TN-039				1.0 Software Engineering					2.0 Software Architecture				3.0 Computing and Exascale			4.0 Unable to Achieve Science Goals for Visibility Data Processing			5.0 Unable to Achieve Science Goals for Non Visibility Data Processing				6.0 Data Products, Data Storage and Data Distribution	7.0 Interfaces for Users and Operators
				1.1 Disassociation between Monitoring and Control and other software-intensive elements of the system	1.2 A wide variety of "antipattern" behaviours	1.3 Misinterpretation and erroneous analysis of requirements	1.4 Limited availability of software engineering resources for exascale	1.5 Limited availability of cross-domain experts	2.1 Lack of attention to non-functional performance requirements	2.2 Overlooking interfaces	2.3 Hardware- specific software architectures	2.4 parallelisation of codes by many orders of magnitude	3.1 Challenges in continuing to scale up the results are not available when needed	3.2 Scaling: sufficiently powerful computers in appropriate price range to deliver science	3.3 MPublished HPC development roadmaps are not delivered	4.1 Obsolescence and technology paradigm shifts	4.2 Algorithm and implementation development; coupled with limited availability of developers	4.3 An enhanced path of continued algorithm development does not deliver performance expectations sufficient to deliver SKA2	5.1 Various imaging-related considerations not yet demonstrated as having been solved	5.2 Implementations are not well specified with large potential impact on processing systems.	5.3 Non visibility data processing	5.4 Too high a level of candidate pulsar detections	6.1 Wide fields of view are required to detect transients	7.1 Definitions of required data products and associated metadata products are late	Human factors are overlooked in developing interfaces
B	Development Environment	1	Development Process	a Repeatability																						
				b Suitability																						
				c Control of Process																						
				d Familiarity with Process or																						
				e Environment Change Control																						
		2	Development System	a Hardware Capacity																						
				b Development System Capability																						
				c Suitability																						
				d Usability																						
				e Familiarity																						
				f Reliability																						
				g Target-Unique System Support																						
				h Security																						
		3	Management Process	a Contingency Planning																						
				b Project Organization																						
				c Management Experience																						
				d Program Interfaces																						
				e Reward Systems																						
		4	Management Methods	a Monitoring																						
				b Personnel Management (Staffing																						
				c Quality Assurance																						
				d Configuration Management																						
		5	Work Environment	a Quality Attitude																						
				b Cooperation																						
				c Communication																						
				d Morale																						
				e Trust																						

S&C Risk Register – links to SEI taxonomy of HPC programmatic risks

WP2-050.020.010-RE-001 / CMU-SEI_2006-TN-039					1.0					2.0				3.0			4.0			5.0				6.0	7.0
					Software Engineering					Software Architecture				Computing and Exascale			Unable to Achieve Science Goals for Visibility Data Processing			Unable to Achieve Science Goals for Non Visibility Data Processing				Data Products, Data Storage and Data Distribution	Interfaces for Users and Operators
					1.1	1.2	1.3	1.4	1.5	2.1	2.2	2.3	2.4	3.1	3.2	3.3	4.1	4.2	4.3	5.1	5.2	5.3	5.4	6.1	7.1
					Disassociation between Monitoring and Control and other software-intensive elements of the system	A wide variety of "antipattern" behaviours	Misinterpretation and erroneous analysis of requirements	Limited availability of software engineering resources for exascale	Limited availability of cross-domain experts	Lack of attention to non-functional performance requirements	Overlooking interfaces	Hardware- specific software architectures	Challenges in continuing to scale up the parallelisation of codes by many orders of magnitude	Scaling: sufficiently powerful computers in appropriate price range to deliver science results are not available when needed	MPublished HPC development roadmaps are not delivered	Obsolescence and technology paradigm shifts	Algorithm and implementation development; coupled with limited availability of developers	An enhanced path of continued algorithm development does not deliver performance expectations sufficient to deliver SKA2	Various imaging-related considerations not yet demonstrated as having been solved	Implementations are not well specified with large potential impact on processing systems.	Non visibility data processing	Too high a level of candidate pulsar detections	Wide fields of view are required to detect transients	Definitions of required data products and associated metadata products are late	Human factors are overlooked in developing interfaces
C	Programmatic	1	Resources	a	Schedule			■	■	■															
				b	Staff	■		■	■	■															
				c	Budget			■							■										
				d	Facilities			■																	
				e	Management Commitment		■																		
		2	Contract	a	Contract Type			■																	
				b	Restrictions		■																		
				c	Dependencies		■																		
		3	Programme Interface	a	Customer Communication		■																		
				b	User Commitment		■																		
				c	Sponsor Alignment		■																		
				d	Subcontractor Alignment		■																		
				e	Prime Contractor		■																		
				f	Corporate Communication		■																		
				g	Vendor Performance		■																		
				h	Political		■																		

Introduction

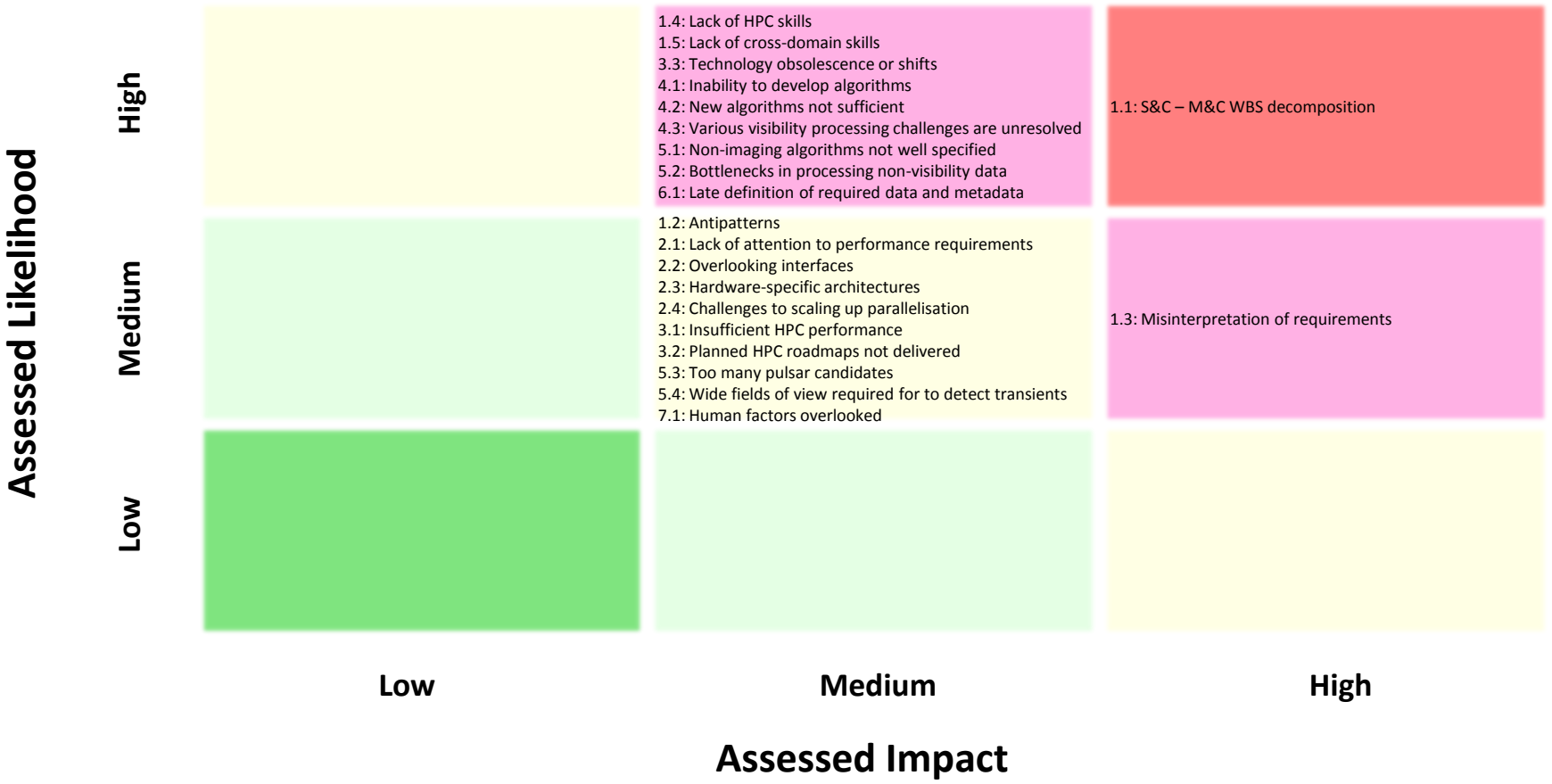
Assessed risk exposures

Comparison with SEI risk taxonomy

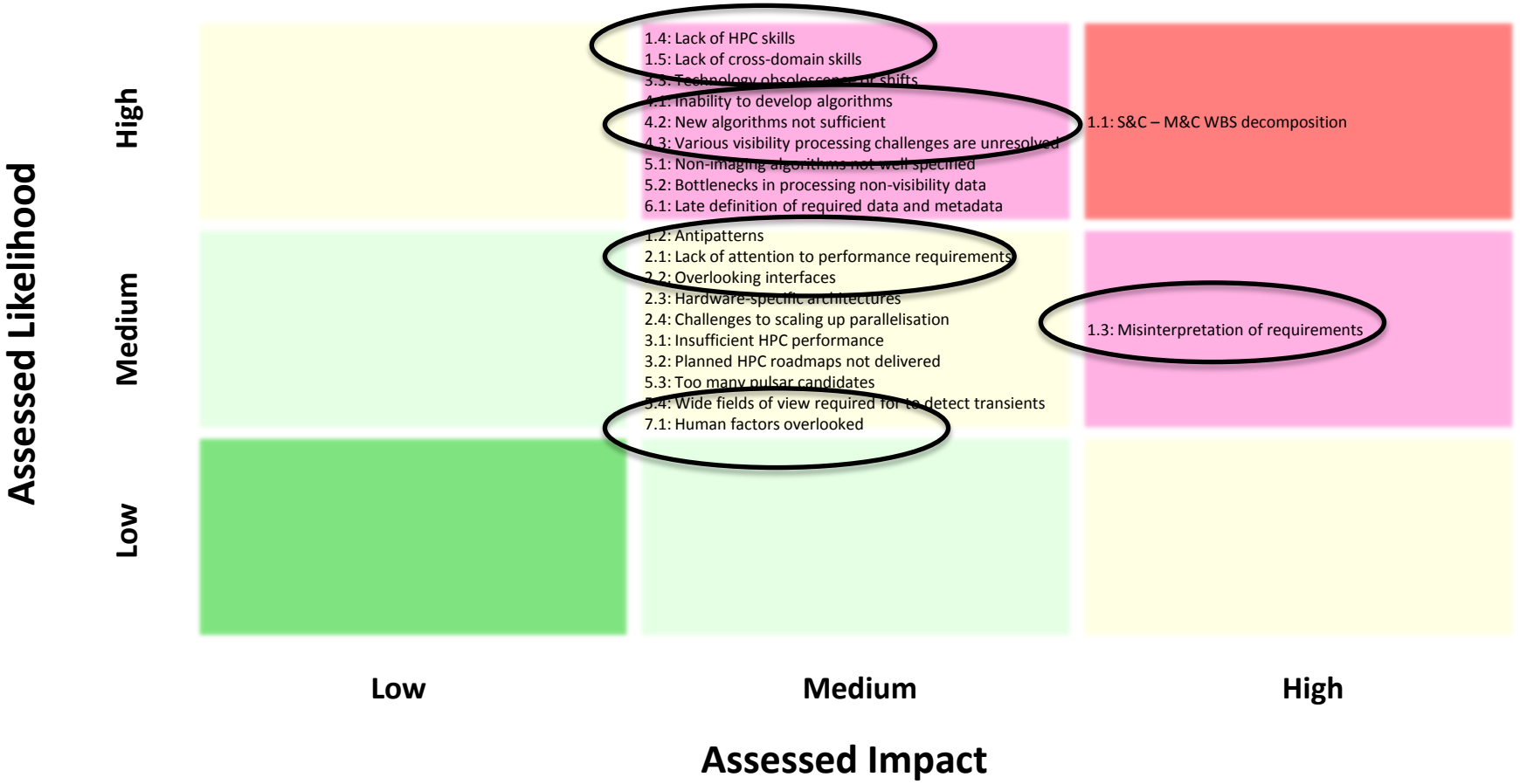
High level analysis

Plans to manage risks and issues

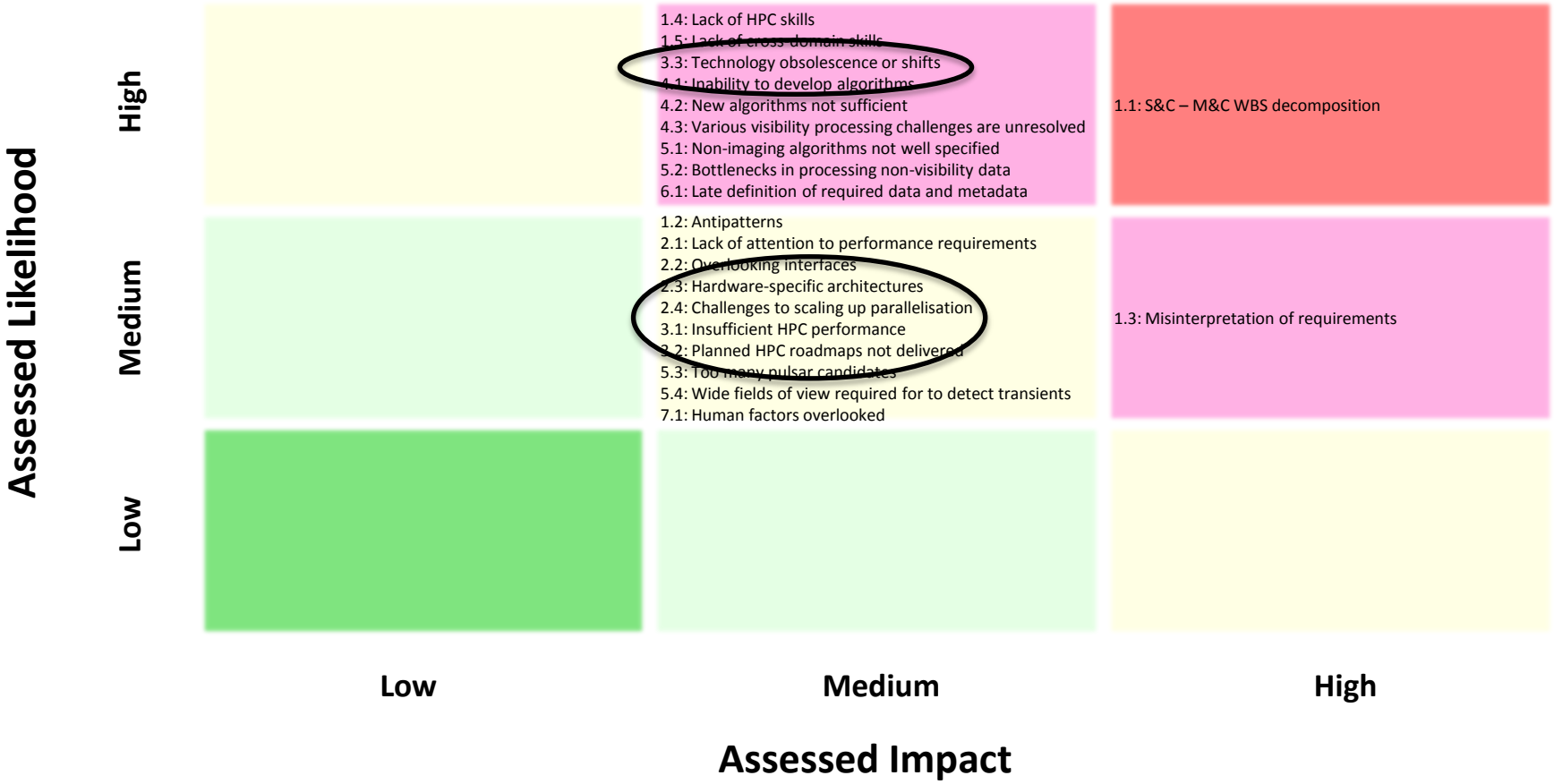
Returning to SKA_1 risks presented in Software and Computing CoDR Risk Register:



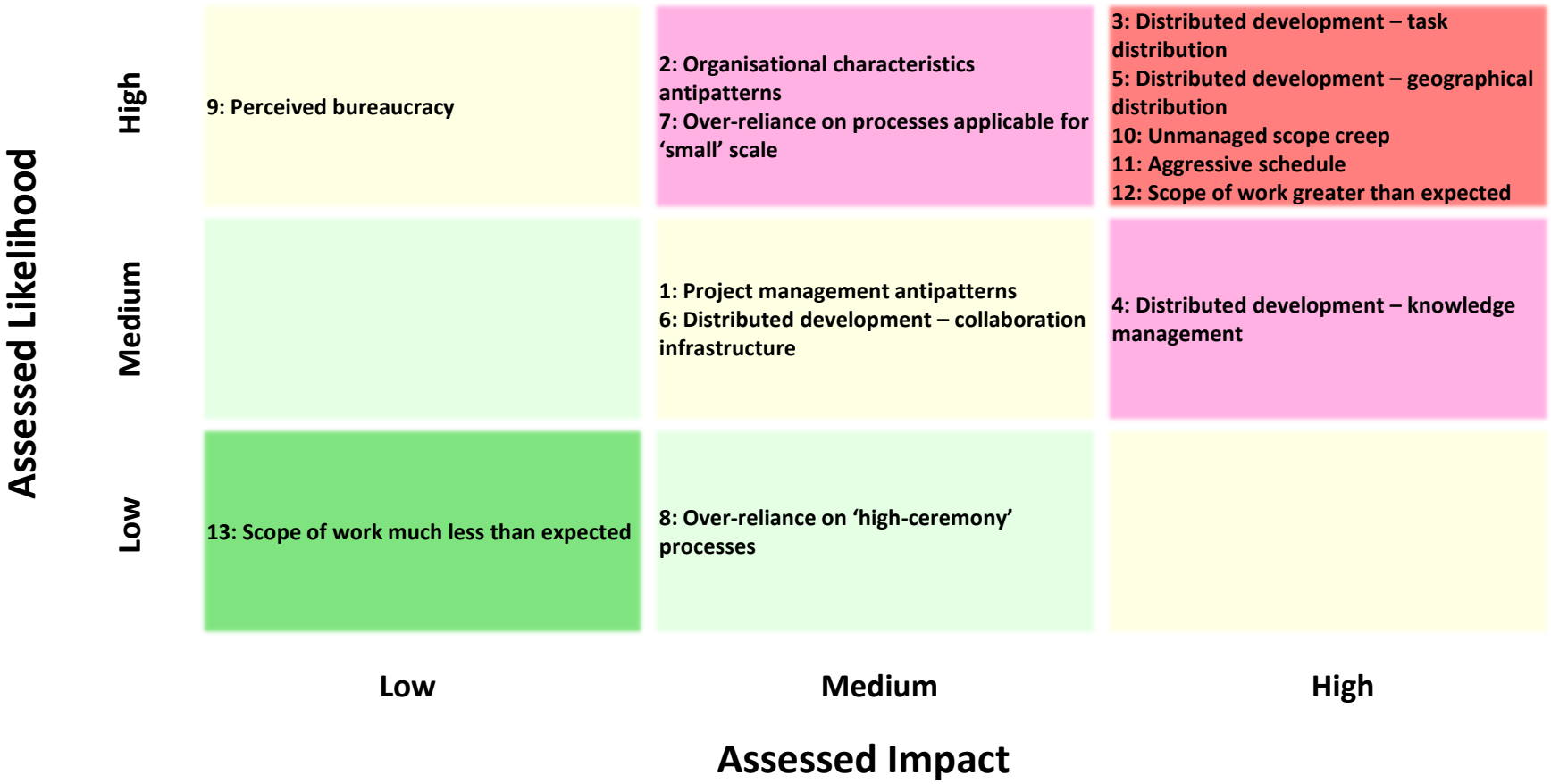
‘People’ related risks are important to manage:



Key ‘technology’ related risks must also be managed:



Many ‘programmatic’ risks have been escalated to System level:



S&C programmatic risks – links to System level risks

			Software and Computing Risks Related to Management and Organisation								
			1.1 A wide variety of "antipattern" project management behaviours	1.2 A wide variety of "antipattern" organisational characteristics	1.3 Distributed Development: Task Distribution	1.4 Distributed Development: Knowledge Management	1.5 Distributed Development: Geographical Distribution	1.6 Distributed Development: Collaboration Infrastructure	1.7 Over-reliance on processes that are generally appropriate only for developing software in the small	1.8 Over-reliance on processes that are generally appropriate only for developing software in the large; coupled with 'perceived bureaucracy' risk	1.9 Perceived bureaucracy
System Level Risks Related to Management and Organisation	Sys_Rsk_1210	Project management systems and controls are ineffective for the SKA project									
	Sys_Rsk_1220	SPDO Contributing Organisations unable to carry out work packages to the point of actual promised delivery.									
	Sys_Rsk_1230	Lack of SPDO staff resources to complete the design and policy work needed for PrepSKA.									
	Sys_Rsk_1240	SKA project structure deficient									
	Sys_Rsk_1250	Loss of valuable experience, relationships and knowledge during project execution and post project.									
	Sys_Rsk_1260	The SKA project fails to understand external project environment									
	Sys_Rsk_1270	Handover between SPDO and SPO									

Introduction

Assessed risk exposures

Comparison with SEI risk taxonomy

High level analysis

Plans to manage risks and issues

Proposed plans to manage risks and issues

- Identify, document and communicate risks:
  - Continuous monitoring, e.g. learning from Pathfinders, Precursors and other projects
  - Health checks – more than just at design reviews
  - Appropriate escalation
- Address issues and high exposure risks:
  - WPCs and SPO must have the skills and processes required to progress work
  - WPCs to actively participate with and learn from industry partners, HPC institutions and collaborations
  - Encourage SPO and WPCs to address cross-cutting concerns via ‘Integrated Design Teams’

END