Designing an Enterprise-wide Governance Framework for Management of Architectural Debt

Min Jeong Yu, 040722

Chair of Software Engineering for Business Information Systems (sebis)
Faculty of Informatics
Technische Universität München
wwwmatthes.in.tum.de

Outline

Motivation

Research Approach

Related Work

Result

I. Expert Interview & Survey

II. Quantification Model

III. Case Study

IV. Governance Framework

Conclusion

Motivation

Why is Architectural Debt (AD) important?

Architectural inadequacy is the most encountered instances of TD.

Holvitie et al., 2014

Architectural aspect of TD has leverage within overall development lifecycle.

Kruchten, 2012

AD concerns the cost of long-term maintenance and evolution of a software system instead of the visible short-term business value.

Kruchten, 2012

Risk when AD makes adding new business value so slow -> widespread refactoring or rebuilding needed

Martini et al., 2014

Lack of quantitative measure / tool to continuously manage AD

Brown et al., 2010

Research Approach

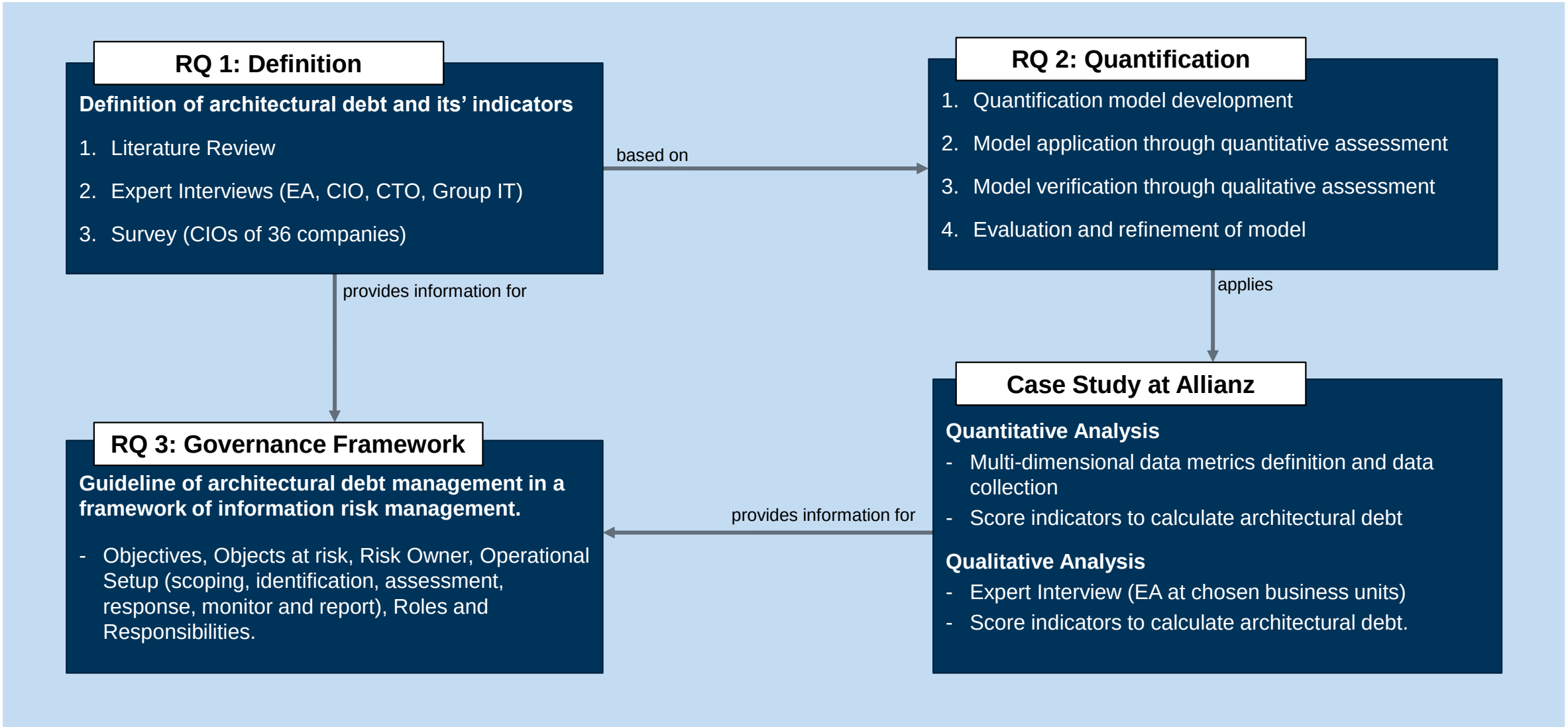
Goal & Research Questions

Title	Designing an Enterprise-wide Governance Framework for Management of Architectural Debt
Goal	Provide guidance on measuring, reducing and governing AD, provide transparency of AD management

Research Questions

- **RQ 1:** What is architectural debt?
- **RQ 2:** How should architectural debt be quantified?
- **RQ 3:** How should business build the governance framework on handling architectural debt?

Research Approach



Literature Review

Existing definitions

Technical debt:
A collection of design or implementation constructs that are expedient in the short term, but set up a technical context that can make a future change more costly or impossible.

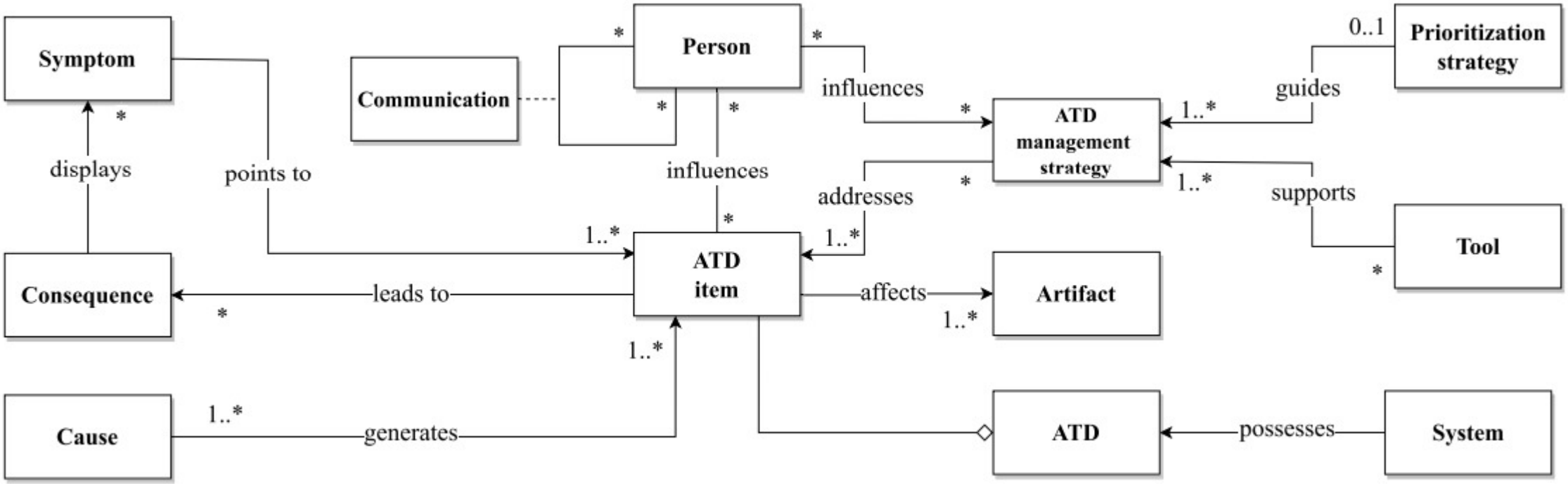
Architectural Debt:
Entirety of the technical debt incurred at the architectural level in a software-intensive system. – Verdecchia

Architectural Debt:
Problems encountered in project architecture, for example, violation of modularity, which can affect architectural requirements. – Alves et al

Enterprise Architecture Debt:
A metric that depicts the deviation of the currently present state of an enterprise from a hypothetical ideal state – Hacks

Architectural Debt Model

Verdecchia explains the relationships and artifacts around architectural debt



Verdecchia 2021

Literature Review

Existing quantification methods

Software Quality Model (ISO 25010)

Product quality model composed of 8 characteristics

- Functionality, Reliability, Usability, Efficiency, Maintainability, Portability

Consequences of Architectural Debt

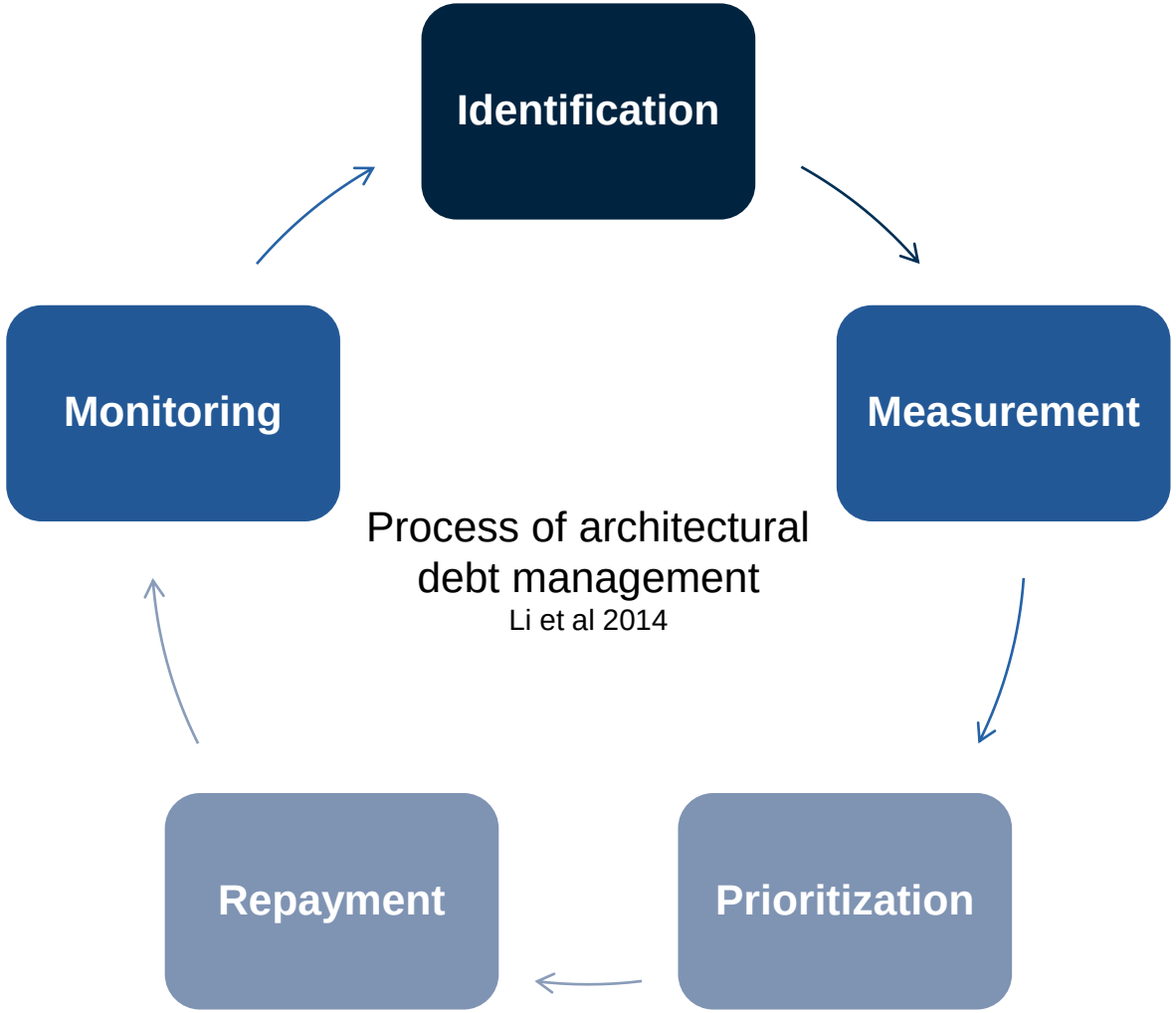
- Maintenance costs, higher evolution - Martini
- Lifecycle properties (understandability, testability, extensionability, reusability) - Mo
- Business- (risk), functionality- (resistance to change), product development-related (difficulties in parallel work) - Verdecchia

Quantification Methods and Tools

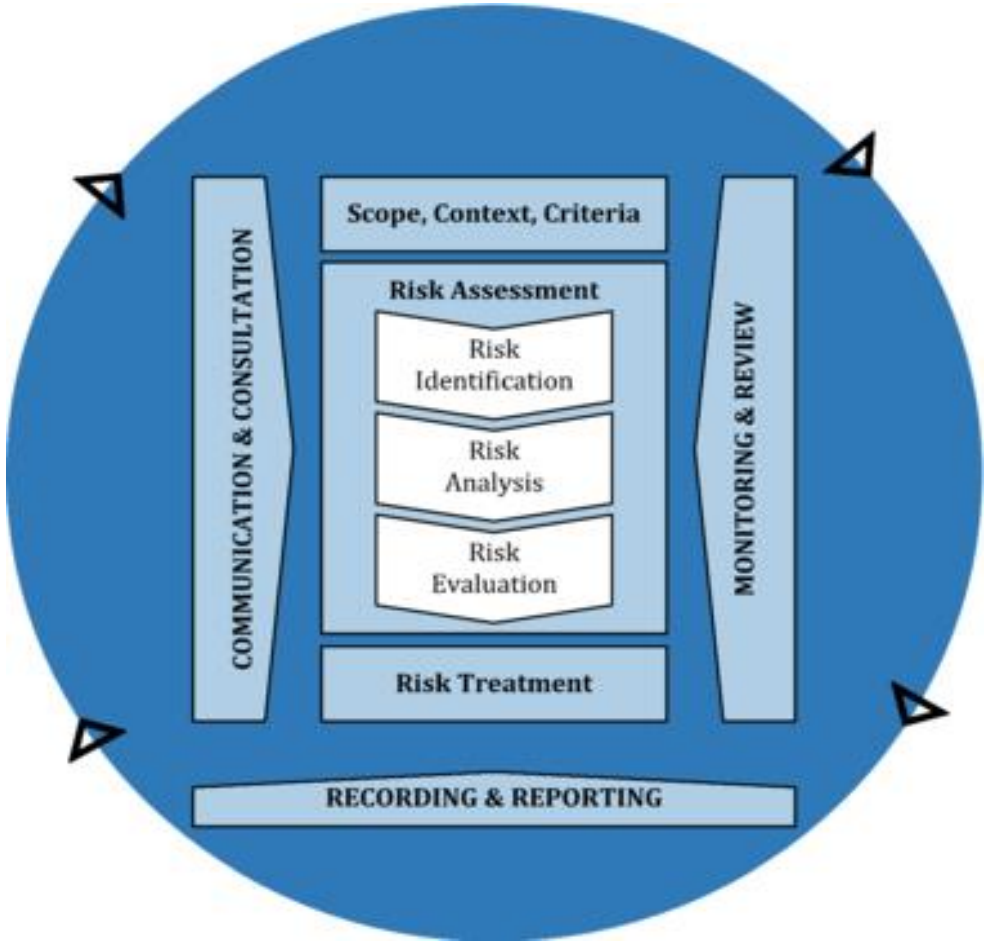
- Technical debt dashboard - Rosser
- Characteristics checklist -Rosser
- Spot evaluation - Rosser
- History coupling probability matrix - Xiao
- Enterprise Architecture Debt: artifact-based framework for business-IT misalignment symptom detection - Hacks

Literature Review

Architectural debt management



Process of architectural debt management
Li et al 2014



ISO31000 Risk Management

Expert Interview and Survey

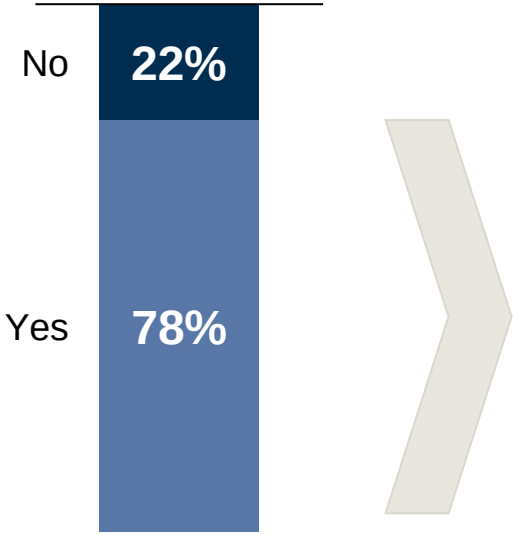
Snapshot of respondents

Interview Respondent	N=6
Title of Respondent	Group CTO at Allianz SE Group CIO at Allianz SE Head of IT Strategy and IT Management at Allianz SE CIO at Allianz Technology SE Enterprise Architect at Allianz SE Consultant at Mckinsey

Survey Respondent	N=32	
Industry	Healthcare, insurance, automotive, tobacco, food, financial services, pharmaceuticals, retail, chemicals, beverages, government	
Region	Americas, Asia and Europe	
Revenues	< 10 billion dollar - +100 billion dollar	
Title of Respondent	group chief architect director enterprise architecture governance enterprise architect IT director head of enterprise architecture senior systems manager regional chief architect	strategic consultant to CIO lead solutions architect VP business applications VP architecture head of strategy principal enterprise architect head of mergers, acquisitions & diversities

Survey Result

Existence of architectural debt management (n= 32)



Respondents' Maturity Level (n=25)

Level 1 19% → Level 2 25% → Level 3 34% → Level 4 16% → Level 5 0%

Q: What is your organization's maturity in identifying, quantifying, tracking, and governing architectural debt of the application landscape?

Indicator for Definition (n=25)

- 80% Complexity of application landscape
- 76% Gap with compliance
- 72% High maintenance Effort
- 60% Age of applications

Q: Which of the following components are included in how your organization defines architectural debt? Please select all that apply.

Metrics for Quantification (n=25)

- 68% Strategic importance of buz. apps
- 64% IT incidents
- 56% Run cost
- 52% Degree of standardization
- 52% Complexity of application landscape

Q: What metrics do you use to measure or quantify your architectural debt? Please select all that apply.

Survey Result

Key challenges and benefits to architectural debt management

Key Challenges (n= 21)

38%	Funding and Budget constraints
24%	Low priority of architectural debt
19%	Resource constraints
19%	Limited business support
19%	Lack of awareness and understanding of the debt
19%	Inability to quantify debt

Key Benefits (n= 20)

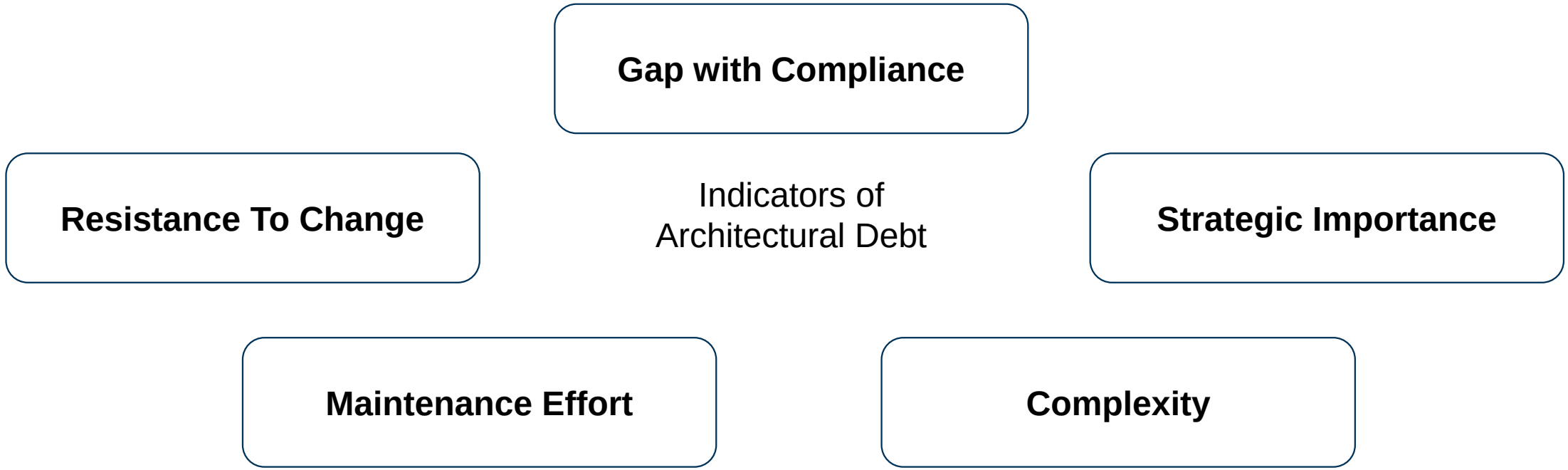
38%	Cost Benefits
24%	Simpler application landscape
19%	Higher agility
19%	Visibility into priorities
19%	Increased efficiency
19%	Improved risk management

Q: What do you believe are the top barriers or challenges to architectural debt management?
Q: What do you believe are the top benefits of a systematic approach to architectural debt management?

Definition of Architectural Debt

Indicators of Architectural Debt

Architectural debt is a term used to explain the gap between the desired target application landscape and the current as-is state of the application landscape due to intentional or unintentional software architectural decisions. On this research, the focus is particularly on the scope of architectural debt of the whole application landscape, instead of a single system/application.



Quantification

Quantification Model Development

1. Define indicators

Indicator 1	Indicator 2

2. Multi-dimensional data metrics

Indicator 1	Indicator 2
Data 1.1	Data 2.1
Data 1.2	Data 2.2

3. Data collection

	Data 1.1	Data 1.2	Data 2.1	Data 2.2
BU 1	X	Xx	Xx	Xx
BU 2	x	xx	xx	xx

4. Rank data (1: good – 3: bad)

	Data 1.1	Data 1.2	Data 2.1	Data 2.2
BU 1	1	2	2	1
BU 2	2	3	1	3

5. Combine data score to rank per indicator

	Indicator 1	Indicator 2
BU 1	1.5	1.5
BU 2	2.5	2

6. Calculate architectural debt risk score (1: very low – 5: very high)

	Average	Arch Debt
BU 1	1.5	**X**
BU 2	2.25	**X**

Case Study – Quantification Model Application

Multi-dimensional Quantification

Resistance to change	Maintenance Effort	Strategic Importance	Complexity	Gap with Compliance
Change Request	IT Incidents	Major Outage	# of Applications in Landscape	Availability: Stability, Breaches
Budget, Run Cost	Reassignments of incidents	Business Value of Business Application	# of Interfaces / Dependencies	Compliance with Group Solutions
Project Duration	Expected Lifetime			
Problem Management				

Case Study – Quantification Model Application

Data used for quantification

Number of Applications	Total number of business applications in an application landscape
Number of Interfaces	Total number of interfaces in an application landscape
Number of Relationship per application	(# flows to + # flows from) / total number of applications Flows to: nr of business applications to which the flow of an application is going Flows from: nr of business applications from which the flow of an application is coming
Incident Resolution Time	Number of incidents resolved in target time / total number of incidents resolved
Incident Response Time	Number of assignments owned in target time / total number of incidents assigned
Major Outage within SLA	Number of major outages with time to repair within KPI target of 4 hours / total number of major outages Major outage: Priority 1 incidents that classifies as having a critical impact
Overall SLA Stability	Number of services/measures with availability KPI met / total number of availability measures
Breaches	Count of instances not fulfilling the stability SLA
Compliance	Number of non-compliant applications / total number of business applications

* SLA: Service Level Agreement

Case Study – Quantification Model Application

Quantitative Analysis Result

1. Calculation of data and level (1: good, 3: bad)

*Size of application is not ranked.

	BU 1	BU 2	BU 3
Size of application	big	big	big
# of interfaces	3	3	3
# Relation per app	3	3	1
Incident Resolution Time	3	2	2
Incident Response Time	1	2	1
Major Outage within SLA	1	3	2
Overall SLA Stability	2	2	1
Breaches	3	3	2
Compliance	3	3	3

2. Combination of levels for each indicator (1: good, 3: bad)

	BU 1	BU 2	BU 3
Complexity	3	3	2
Maintenance	2	2	1,5
Strategic importance	1	3	2
Gap with Compliance	2,667	2,667	2
Average	2,1667	2,667	1,875

3. Calculation of architectural debt risk score(1: very low, 5: very high)

	BU 1	BU 2	BU 3
Overall Architectural Debt	3	5	3

Case Study – Quantification Model Verification

Qualitative Analysis Result

	Quantitative Assessment Result	[BU 3] Interview 1	[BU 3] Interview 2
Title of Interviewee		Executive Enterprise Architect	Application Governance Architect
Assessment (1:excellent 5: poor)	Resistance to Change ---------- x Maintenance Effort ---------- 2 Strategic Importance ---------- 3 Complexity ---------- 3 Gap with Compliance ---------- 3	Resistance to Change ---------- 4 Maintenance Effort ---------- 2 Strategic Importance ---------- 3 Complexity ---------- 4 Gap with Compliance ---------- 3	Resistance to Change ---------- 4 Maintenance Effort ---------- 4 Strategic Importance ---------- 1 Complexity ---------- 5 Gap with Compliance ---------- 3
Remarks		Clustering of global / local operation	Major outage cannot represent strategic importance alone. (gap with target landscape)

- Assessments correspond to the results from the quantitative assessment result.
- Justifications and judgment on scores were given. Ex) due to global operation and history of combining different business units to the current application landscape, complexity is high.

Governance Framework

***Governance Framework** includes:*

Objectives, definition of objects at risk and risk ownership, roles and responsibilities, and operational setup of architectural debt management

1. Objectives

Provide an integrated method, process and supporting tool throughout an enterprise. This ensures that architectural debt is consistently documented and managed across business units.

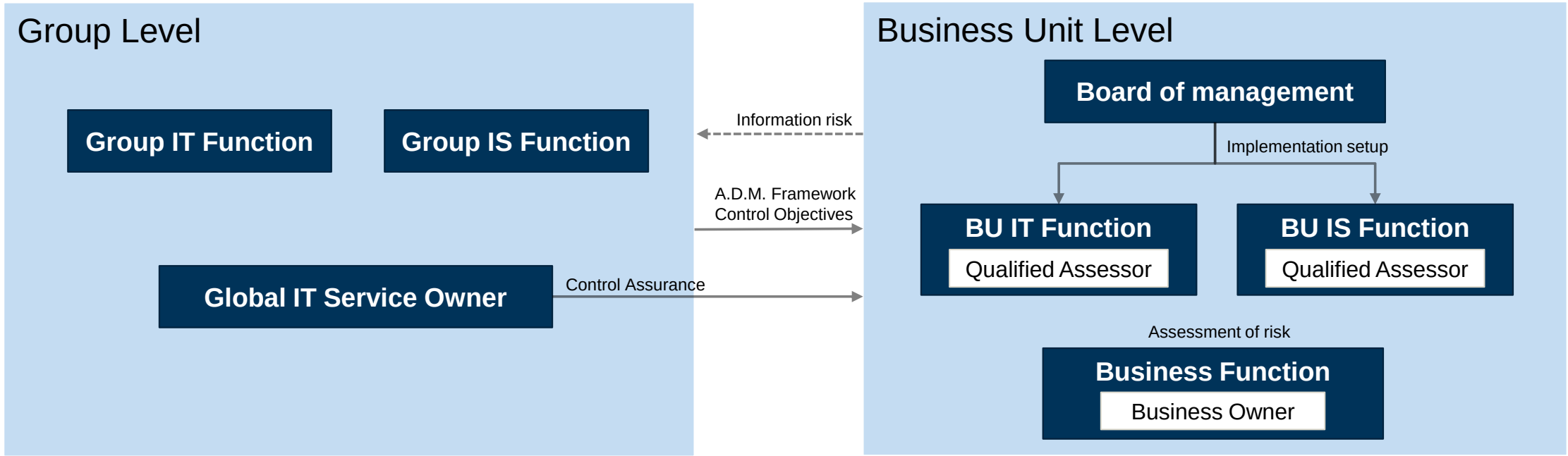
2. Define objects at risk and risk ownership

Information Risk Owner	Object at Risk
Business Owner	Application Landscape [Primary]
Business Owner	Business Applications [Secondary]

Governance Framework

3. Roles and Responsibilities

- **Group IT&IS**: overall responsibility
- **Global IT service owner**: assessing inherent risk associated to their mandatory shared IT service
- **Board of management** at business unit: implementation of organizational and operational setup to comply to governance framework
- **Qualified assessors**: assessment of information risks within their area of responsibility
- **Business owners**: risk owners that assess risk for their business applications and ensure mitigations to achieve acceptable level of residual risk.



Governance Framework

4. Operational Setup

i. **Scoping**: prioritization of objects at risk for regular testing of controls

ii. **Identification**: continuous process to identify architectural debt

iii. **Assessment**: architectural debt is calculated and qualified assessor verifies the score

iv. **Response**: highly migrate | mitigate risk | accept risk

v. **Monitoring & Reporting**: track implementation and report to relevant stakeholder. Re-validation of architectural debt based on severity level.

Field	Type	Description
Object at risk	Reference	Primary objects at risk
Qualified assessor	Person	Person, who assesses the risk
Quantitative Assessment Score	Enum	Rating of architectural debt risk score acc. to quantitative assessment (1: very low – 5: very high)
Qualitative Assessment Score	Enum	Rating of architectural debt risk score acc. to judgment of qualified assessor (1: very low – 5: very high)
Data privacy impact	Bool	Indicator showing whether a data privacy impact is associated with risk or not (TRUE – FALSE)

Information Risk Documentation

Risk Migration Plan	Type	Description
Information risk	reference	Assessed Information Risk that is to be highly migrated
Confirmation	person	Stakeholders that confirms the migration/replace plan
Start date	date	Date at which the forward planning is started
End date	date	Date at which the forward planning is planned to be finished
Progress	enum	Monthly confirmations for the progress of the plan (on track – delayed)

Risk Migration/Replace Plan

Conclusion

Limitation and Outlook

Limitation

Quantification Model

- Manual quantification system
- Impact–based identification, rather than cause–based. May contain other factors than architectural debt

Case Study

- Different data quality and operating system among business units leading biased ranking result
- Limited control and access on data
- Qualitative Assessment may be biased due to different knowledge/expertise level of interviewees.

Outlook

Quantification Model

- Automated quantification system
- Monetary value quantification (principal, interest) for communication to business

Case Study

- Expansion and enhancement of metrics and indicators
- Further analysis on data like clustering size and characteristics of application landscape

Appendix

Appendix – Result

RQ 1: Definition

Definition of architectural debt and its' indicators

Definition: the gap between the desired target application landscape and the current as-is state of the application landscape due to intentional or unintentional software architectural decisions.

Indicators: Complexity, Strategic Importance, Gap with Compliance, Maintenance Effort, Resistance to Change

based on

RQ 2: Quantification

Quantification Model

1. Multi-dimensional data based on indicators defined: change request, IT Incidents, project duration, cost, lifetime, availability, size, interfaces, major outage, architecture exceptions request
2. Data collection of application landscapes
3. Rank in 3 levels for each indicator
4. Calculate architectural debt acc to levels scored

+

RQ 2: Quantification

Qualitative Analysis

1. Expert Interview (EA): Score indicators in levels to calculate architectural debt.
2. Verify results from quantitative quantification.

provides information for

RQ 3: Governance Framework

Guideline of architectural debt management in a framework of information risk management.

Objects at risk: Business applications in application landscape

Risk Owner: Business owner at BU

Roles and Responsibilities: Qualified assessors of BU

Operational Setup
- Scoping: Prioritize objects at risk
- Identification: Data collection of indicators that provide risk.
- Assessment: Quantification model that analyses data.
- Response: Decision of Highly migrate | Mitigate | Accept
- Monitor & Report: Re-validate risk in line with risk level

Appendix – Survey Plan

Action	Actors
Initial agreement on content and plan	conductor, research board
First draft of questionnaire	conductor
Review and final draft	conductor, editor from research board
Broadcast survey	research board
Respond to survey – 3 weeks	respondents
Collection of responses	conductor, research board
Analysis and conclusion	conductor

Appendix – Quantification of Architectural Debt

Qualitative Assessment through Expert Interviews

Resistance to change	Maintenance Effort	Strategic Importance	Complexity	Gap with Compliance
Change request, Project duration, Budget	IT Incidents, Lifetime	Major Outage	Size of landscape, Dependencies / Interfaces	Availability, Architecture Exceptions Requests

Assess your OE's architectural debt in the application landscape for each indicator above.

Level 1 – Optimized
Level 2 – Good
Level 3 – Fair
Level 4 – Poor
Level 5 – Non-existent

Appendix – Risk Assessment Steps

Assessment Steps	Rating	Example
1. Collection of data and level per metrics	(1: good, 3: bad)	
2. Combination of levels for each indicator	(1: good, 3: bad)	
3. Architectural debt risk score	(1: very low, 5: very high)	
4. Verify score by judging indicators and calculating the score in qualitative assessment	(1: very low, 5: very high)	Resistance to Change ---------- x Maintenance Effort ----------x Strategic Importance ---------- x Complexity ---------- x Gap with Compliance ---------- x **Architectural Debt Score** ---------- x

	BU 1	BU 2	BU 3
Size of application	big	big	big
# of interfaces	3	3	3
# Relation per app	3	3	1
Incident Resolution Time	3	2	2
Incident Response Time	1	2	1
Major Outage within SLA	1	3	2
Overall SLA Stability	2	2	1
Breaches	3	3	2
Compliance	3	3	3

	BU 1	BU 2	BU 3
Complexity	3	3	2
Maintenance	2	2	1,5
Strategic importance	1	3	2
Gap with Compliance	2,667	2,667	2
Average	2,1667	2,667	1,875

	BU 1	BU 2	BU 3
Overall Architectural Debt	3	5	3

Appendix – Information Risk Information

Field	Type	Description
Object at risk	Reference	Primary objects at risk
Qualified assessor	Person	Person, who assesses the risk
Quantitative Assessment Score	Enum	Rating of architectural debt risk score acc. to quantitative assessment (1: very low – 5: very high)
Qualitative Assessment Score	Enum	Rating of architectural debt risk score acc. to judgment of qualified assessor (1: very low – 5: very high)
Data privacy impact	Bool	Indicator showing whether a data privacy impact is associated with risk or not (TRUE – FALSE)

Appendix – Risk Migration Plan

Risk Migration Plan	Type	Description
Information risk	reference	Assessed Information Risk that is to be highly migrated
Confirmation	person	Stakeholders that confirms the migration/replace plan
Start date	date	Date at which the forward planning is started
End date	date	Date at which the forward planning is planned to be finished
Progress	enum	Monthly confirmations for the progress of the plan (on track – delayed)

Appendix – Risk Mitigation Plan & Acceptance Information

Risk Mitigation Plan	Type	Description
Information risk	reference	Assessed Information Risk that is to be mitigated
Confirmation	person	Stakeholders that confirms the mitigation plan
Start date	date	Date at which the mitigation actions are started
End date	date	Date at which the mitigation actions are planned to be finished
Progress	enum	Monthly confirmations for the progress of the plan (on track – delayed)

Risk Acceptance	Type	Description
Information risk	reference	Assessed Information Risk that is accepted
Confirmation	person	Stakeholders that confirms the risk acceptance
Expiration date	date	Date at which the risk acceptance is due to re-validation