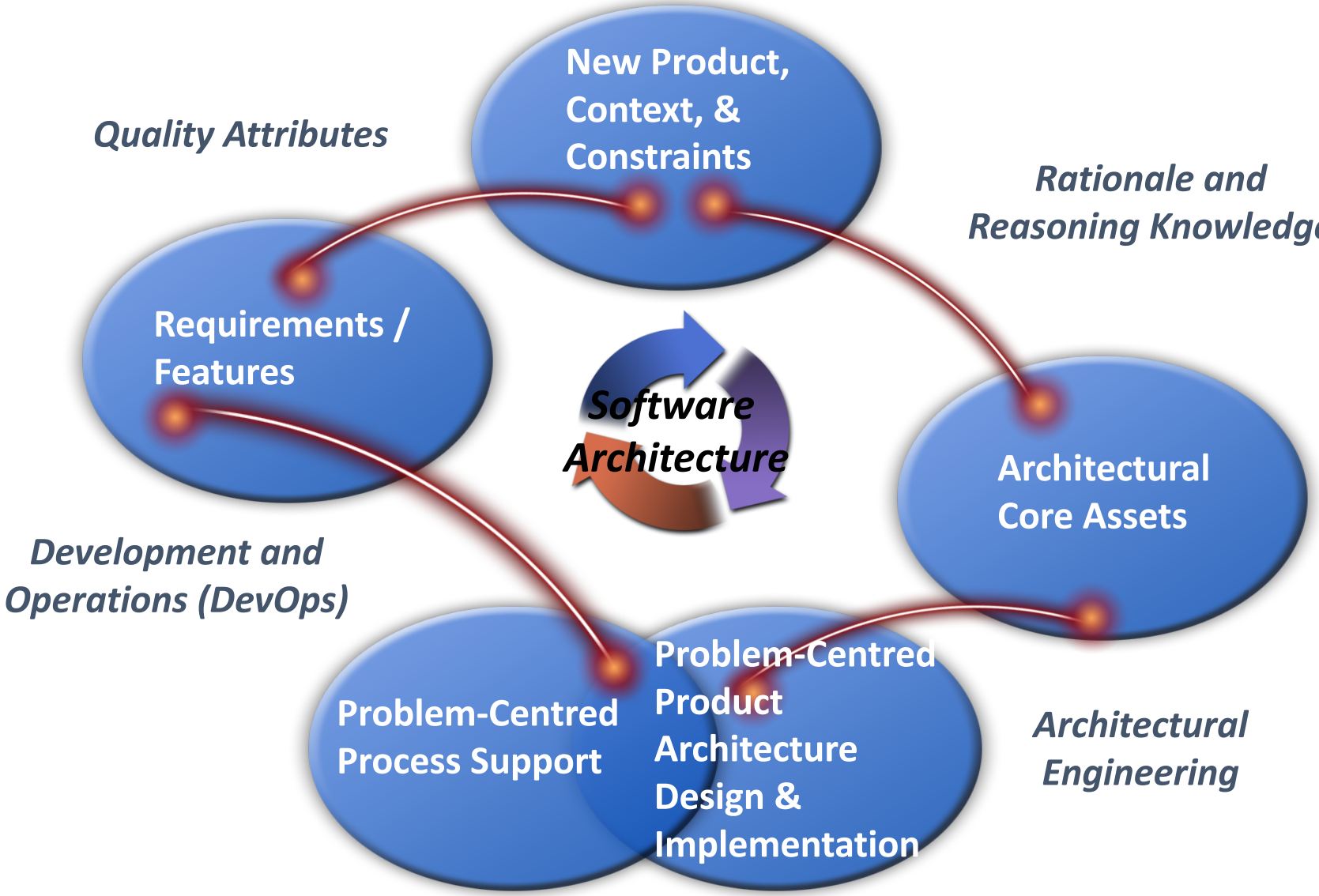
Architecture and DevOps

M. Ali Babar

CREST – Centre for Research on Engineering Software Technologies
University of Adelaide, Australia

Architecture-Centric Research for Software Services



The Key Areas of the Current Projects

Science of Cyber Security

- Methods and tools for evidence synthesis and evaluation
- Mining Software Repositories for security knowledge
- Field Studies for developing and deploying secure Software using DevOps paradigm and tools

Secure and Scalable Private Cloud Infrastructures

- Design and implementation of secure private cloud
- Semi-automated configuration of tools/ pipelines
- Integration and testing of Docker, kubernetes, rkt, and LXD for secure private cloud deployments

Resilient Architectures for Cloudlet/Edge computing

- Building and leveraging resilient architectures for Cloud systems, Internet of Things (IoT) and Fog Computing
- Formal qualitative and quantitative resilience metrics development and evaluation

Data Exfiltration and Security Orchestration

- Strategies for uncovering software flaws for data exfiltration
- Designing and deploying adaptive countermeasures
- Architectural support for automatic security orchestration

Cyber Security CRC – Program Structure

2 programs of research will meet infrastructure, platform architecture and software challenges and opportunities relating to cyber security.

Program	Blurb	Project	Description
1 Critical Infrastructure Protection	Mitigating threats from nation states, criminals, and other human use factors	**1.1 Resilient Networks**	Develop new techniques and technologies for: reliably capturing and filtering high-speed, high-volume network traffic; detecting and diagnosing anomalies in such traffic; remotely interrogating the configuration of "rogue" devices; providing standardised data sets for testing new network protection technologies; incorporating cryptographic and authentication solutions into control networks; and on-the-fly verification of data integrity and authenticity
		1.2 Security & configuration management of IoT systems	The project will provide risk assessment, evaluate new threats, and develop new platforms/technologies to provide secure deployment and management of IoT/IoE, end user networked devices and machine-machine (M2M) networks anywhere/anytime.
		1.3 Development of a national authentication system	This project will develop a national authentication system that can be used in an e-government context or commercial situation. The system will enhance privacy and trust for Australians by building on existing national systems used in other countries, e.g. Estonia, New Zealand.
		1.4 Forensics and responses to emerging threats	Produce systems to ensure forensic level traceability of cyber security incidents, including across systems of systems as represented in many critical infrastructures. The project's outcomes will use AI principles to generate operational efficiencies through forensically validated automation of currently manual operational tasks
2 Cyber Security Solutions as a Service	Ensuring solutions that are scalable, broadly applicable, and leverage machine learning techniques to ensure process improvement	**2.1 Platform and architecture for cyber security as a service**	This project will provide a secure integrated platform for providing security as a service. The platform will be based on a reference architecture designed and evaluated using security-by design paradigm
		2.2 Security automation and orchestration	Security automation and orchestration have become imperative. This project aims to provide advanced technologies for security orchestration by developing self-adaptation and self-healing tools that can be provisioned as SaaS
		2.3 Privacy preserving data sharing in a hyperconnected world	Data processing for security needs, access to and analysis of fine-grained data without compromising privacy. This project will provide enabling knowledge and tools for privacy preserved analysis of data for security
		2.4 Real time monitoring of cyber threats	Produces a system of systems that through fusion of advanced real time monitoring of cyber security threats enables situational awareness of cyber threat and risk through advanced visualisation techniques

Evidence-Based Approach to Exploring the Relation between Architecture and DevOps Paradigm

(Re)-Architecting for DevOps

- ***(Re-) architecting to enable continuous delivery and deployment?***
  - Three challenges: *highly coupled monolithic architecture*, *team dependencies*, and *ever-changing and complex environments*.
  - Six principles: *small and independent deployment units*, *not too much focus on reusability*, *aggregating logs, supporting frequent and incremental changes*, *design for failure*, and *testability inside an architecture*.
  - Autonomy and decomposition strategies: *deployability*, *modifiability*, *testability*, *scalability,* and *team-scale*
- Designing highly operaionalised architectures: *prioritise operational concerns early, continuously engage with the operations staff*, and *leverage logs and metric data for operational tasks*.

(Re)-Architecting for DevOps

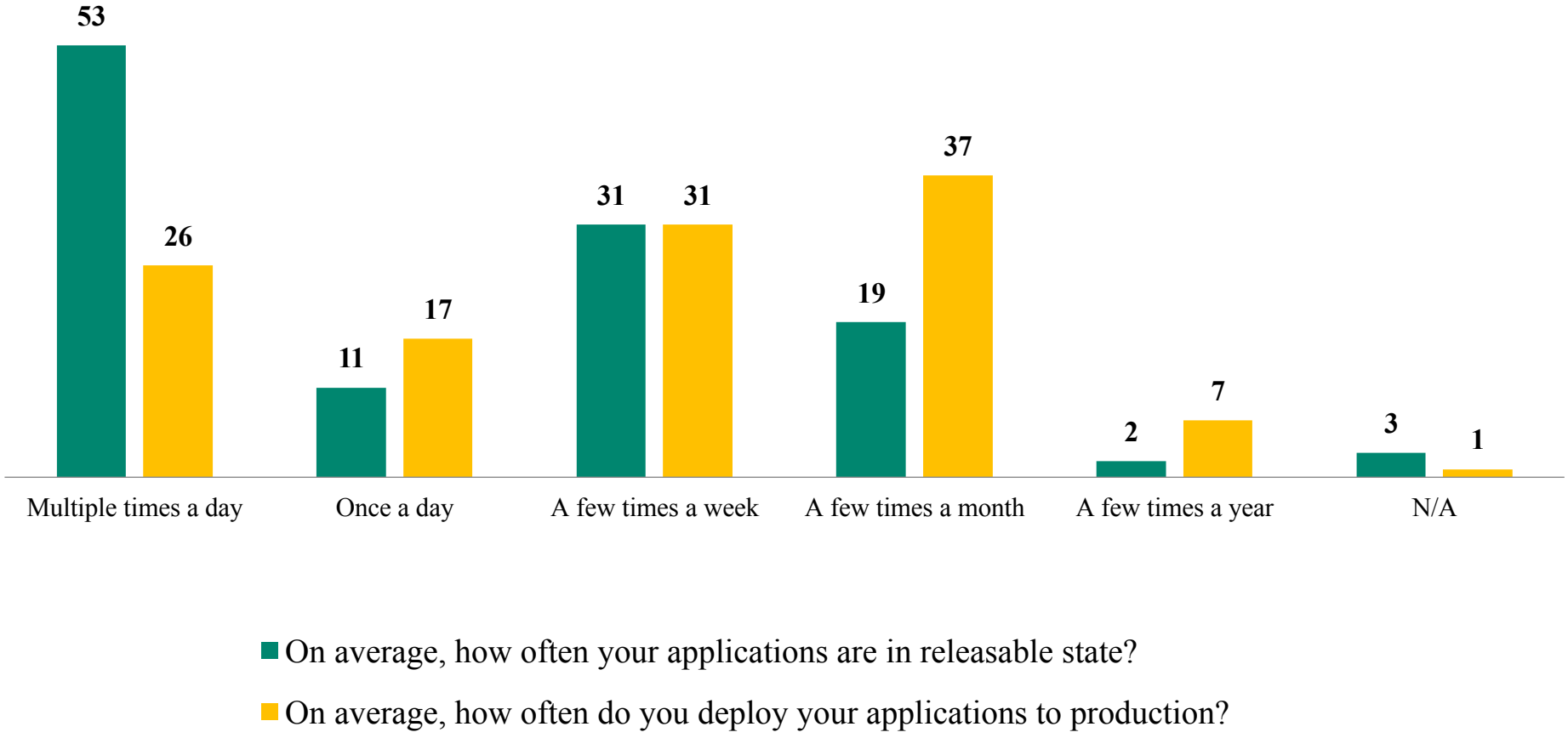
- ***Contributions of the research so far!***

  - *A better understanding of practicing CD within monoliths and identifies a list of reasons for disruptions to CD adoption within the monoliths by exploring the practitioners' perceptions;*
  - *A characterization of "small and independent deployment units" principle attempted by the participants to ease a CD journey;*
  - *A set of quality attributes that require more attention when designing an application in CD context;*
  - *An empirical evidence about the perceived benefits of addressing operational aspects during architecting phase of an operations-friendly architecture;*
  - *A catalogue of findings about architecting for CD that can be used as guidance for further research effort and provide concrete recommendations for better practices and tools development.*

Findings From Surveyed Questions

Continuous Delivery vs. Deployment in practice

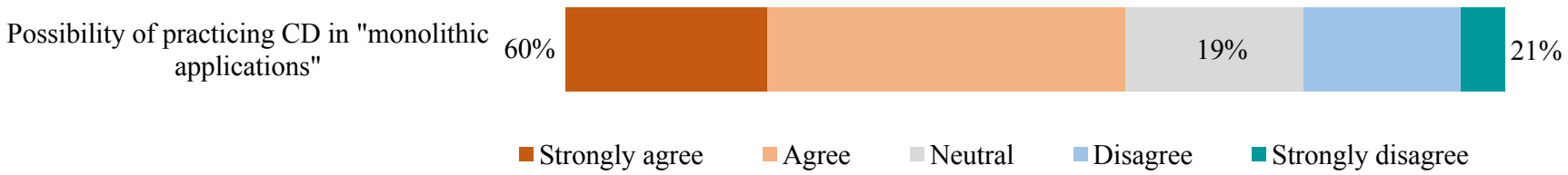
Finding 1: From a practitioner's perspective, continuous delivery and continuous deployment are indeed distinguishable practices in industry.



Monoliths and CD

Finding 2: Monoliths and CD are not intrinsically oxymoron.

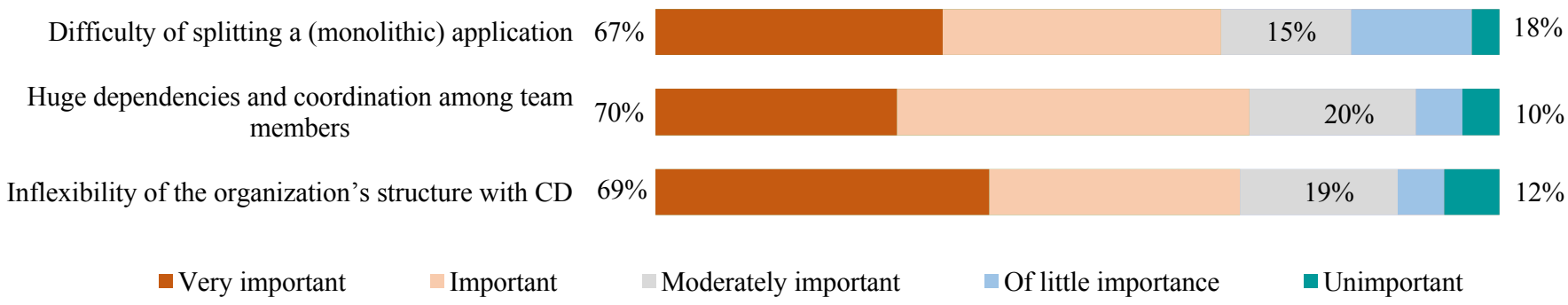
Finding 3: Adopting CD in monoliths is more difficult, as there are hurdles for having team autonomy, fast and quick feedback, enabling automation (e.g., test automation) and scalable deployment.



Monoliths and CD

Finding 4: Breaking down monoliths into smaller pieces brings more flexibility in CD; however, the participants experienced it as challenging process.

Finding 5: Inflexibility of organizational structure (e.g., team structure) with the spirit of CD is the most critical challenge for implementing CD.



Moving Beyond Monoliths

Finding 6: "Small and independent deployment units" is a key principle, which is widely used as an alternative to monoliths, and serves as a foundation to design CD-driven architectures.

Finding 7: Autonomy in terms of deployability, modifiability, testability, scalability, and isolation of business domain are the main characteristics of this principle.

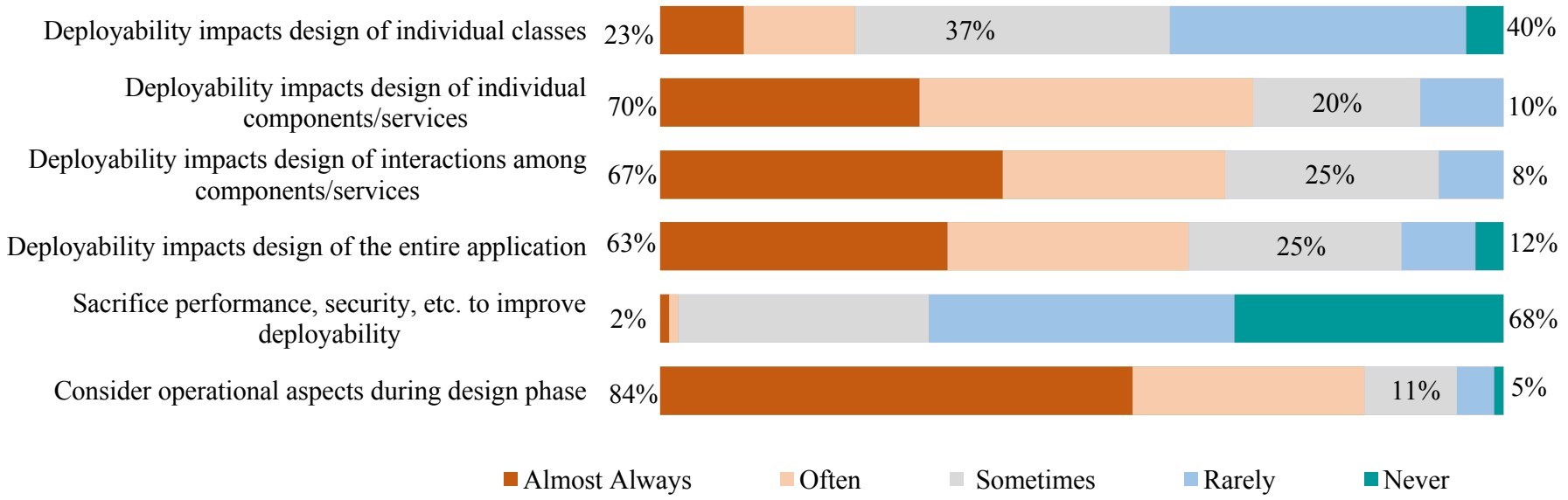
Finding 8: Adopting microservices to promote delivery speed comes at a cost as it necessitates considering organizational structures and highly skilled team. Ignoring this fact may negatively impact the deployment capability of an organization.

Quality Attributes that Matter in CD

Deployability
Testability
Modifiability
Monitorability/Loggability
Resilience
Reusability

Deployability

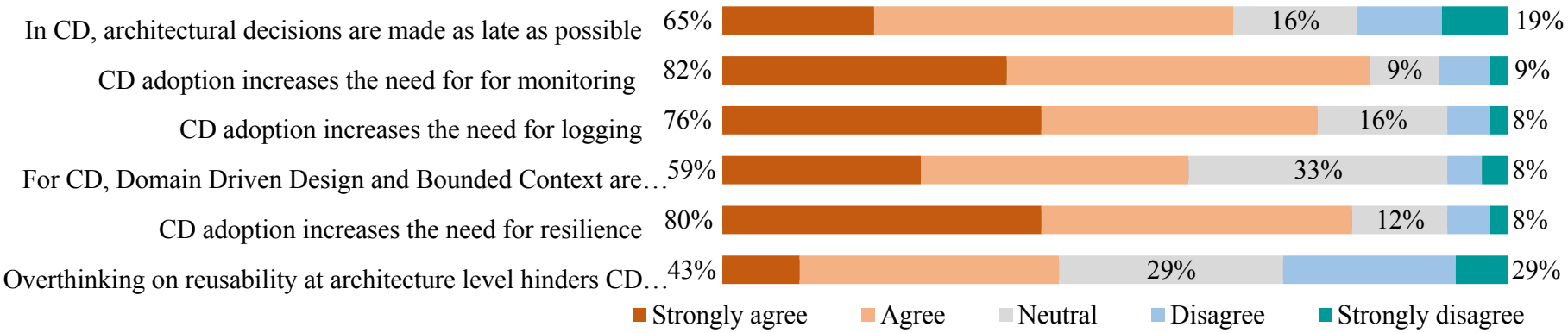
Finding 9: Concerns about deployability impact how applications are designed, however interactions among components/services are most influenced by deployability.



Quality Attributes that Matters (Largely/Less)

Finding 10: The importance of monitorability, loggability and resilience has increased, but overthinking about "reusability" at architecture level may negatively impact CD adoption.

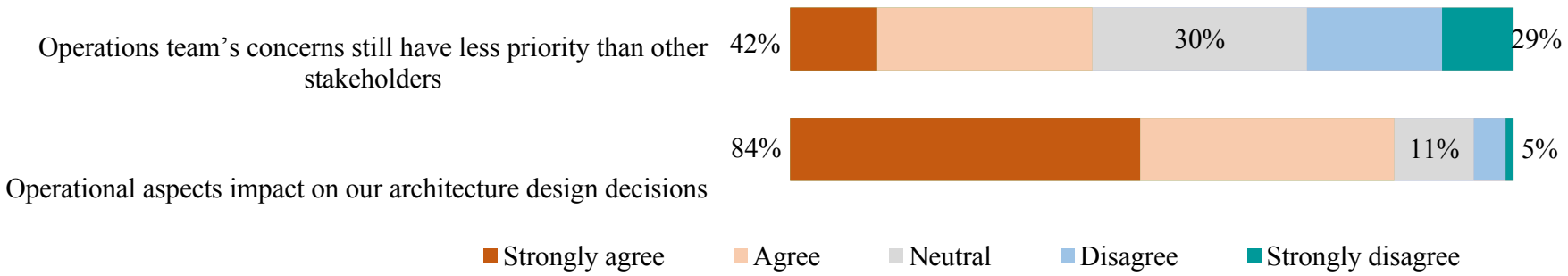
Finding 11: Compared to less frequent release, CD more emphasizes on evolutionarily changes. This requires delaying architectural decisions to the last possible moment.



Operation Aspects and Architecture

Finding 12: Considering operational aspects early in software development process would help design and implement operations-friendly architectures.

Finding 13: CD can expand architect's role as apart from software design they need to deal with infrastructure architecture, test architecture and automation.



Acknowledgement

- The empirical study is the part of Mojtaba Shahin PhD research

- The Cyber Security CRC program has multiple academic and dozens of industrial and governmental agencies

Thank You!

Questions