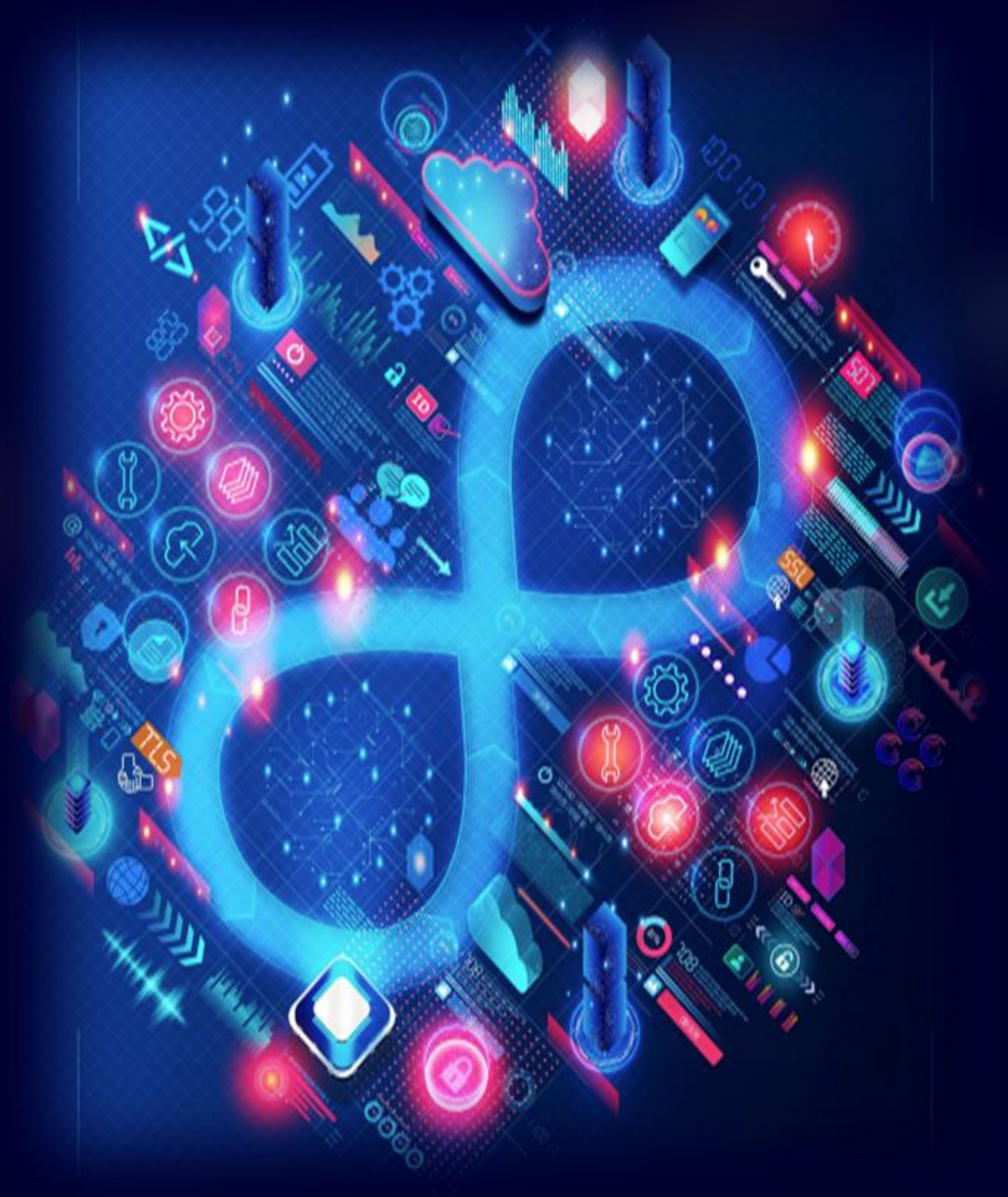
Secure Software Development Life Cycle Plan for SnowBe



Framework Selection: NIST SSDF

Comprehensive Approach

Integrates security throughout the entire software lifecycle, not as an afterthought.

Flexibility

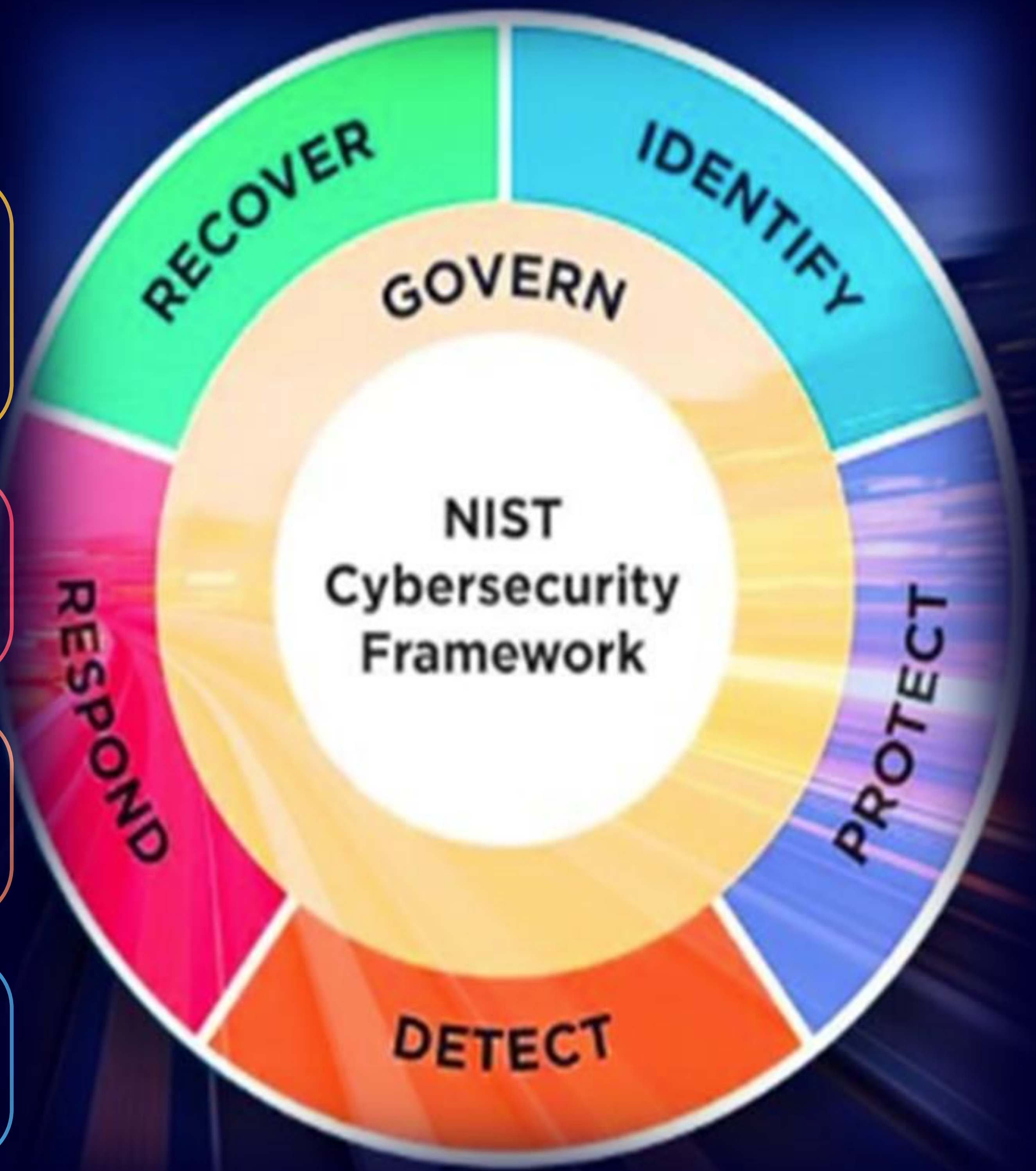
Adaptable to SnowBe's evolving needs as a growing company.

Compliance Support

Aligns with PCI DSS requirements for credit card transactions.

Supply Chain Security

Addresses CEO's concerns about third-party risks.



Decision Process

Security Needs Assessment

Identified SnowBe's lack of existing processes and sensitive data storage requirements.

Framework Comparison

Evaluated Microsoft SDL against NIST SSDF for SnowBe's specific context.

Compliance Mapping

Confirmed SSDF aligns with NIST 800-53 r5 and PCI DSS requirements.

Implementation Planning

Assessed ease of adoption for a new development team.

Why Not Microsoft SDL?

Assumes Existing Practices

Requires developers to already follow security best practices, which SnowBe lacks.

Enterprise Focus

Designed for large-scale development with strict guidelines.

Windows Ecosystem

Less adaptable to SnowBe's AWS, WordPress, and web technologies.

Limited Supply Chain Focus

Doesn't address supply chain risks as comprehensively as SSDF.

Karen: Project Manager & Security Lead

Oversees development processes, ensures SSDF compliance, and works with executives on business goals.

Developer 1: Frontend Engineer

Builds secure user interfaces, implements input validation, and secure authentication.

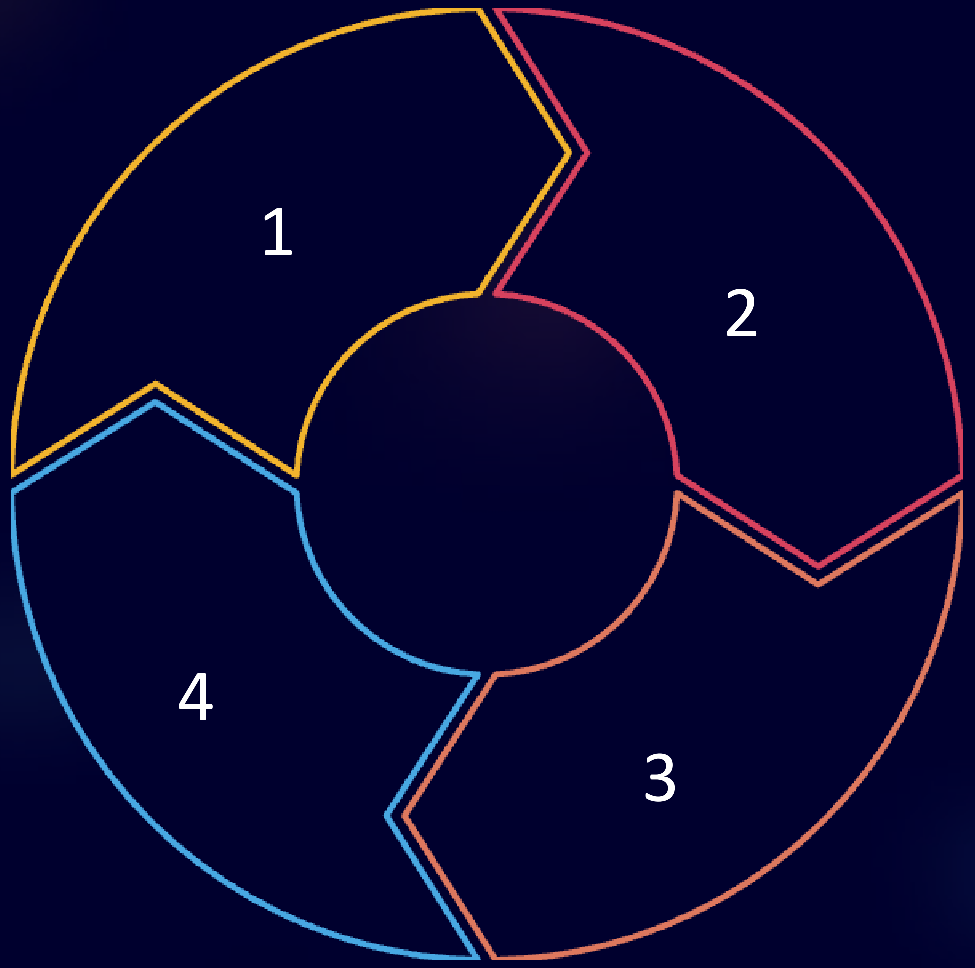
Developer 2: Backend Engineer

Develops server-side logic, prevents SQL injection, and implements access controls.

Developer 3: DevOps & Security

Manages CI/CD pipeline security, automated testing, and secure deployment practices.

Development Methodology



Agile Practices

Enables quick response to business needs.

DevSecOps Integration

Embeds security throughout the development lifecycle.

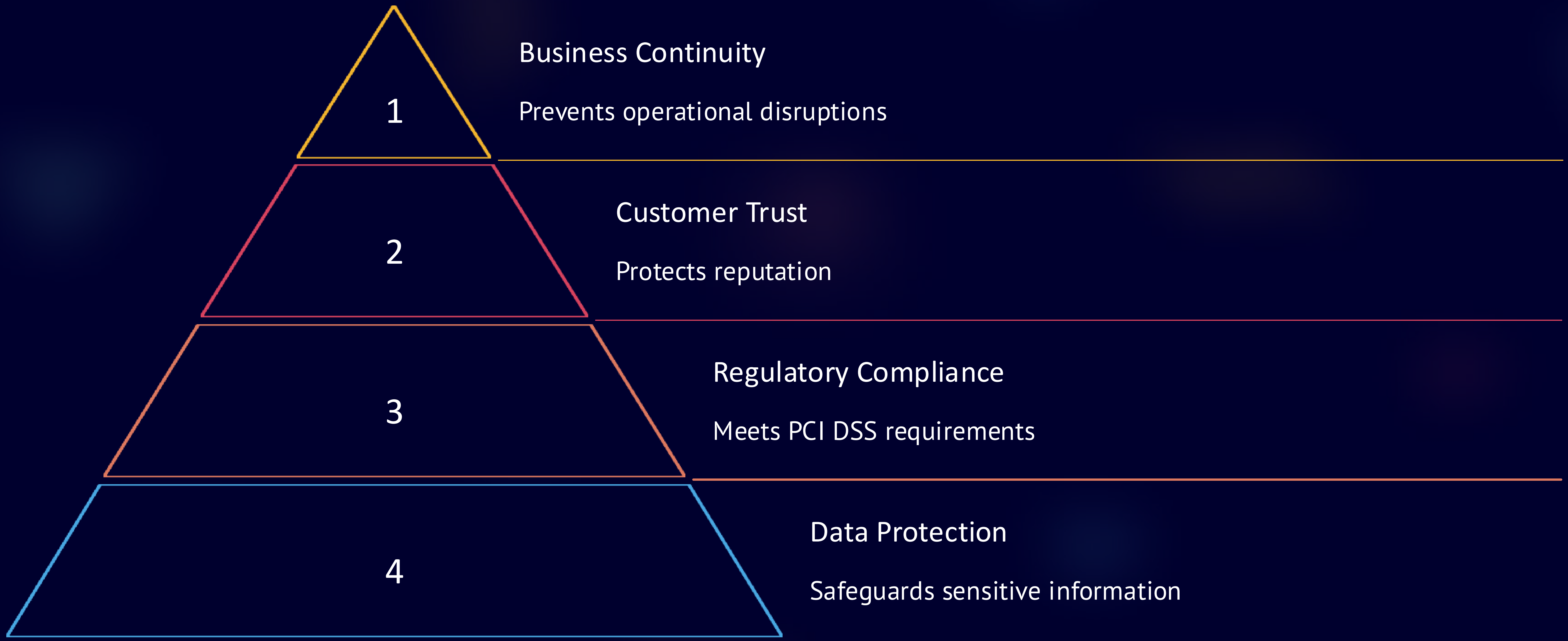
Automated Security

Reduces risk of misconfigurations and code flaws.

Cross-functional Collaboration Collaboration

Promotes team efficiency and knowledge sharing.

Importance of Secure Software



Business Continuity

Prevents operational disruptions

Customer Trust

Protects reputation

Regulatory Compliance

Meets PCI DSS requirements

Data Protection

Safeguards sensitive information

Implementation Roadmap

1

Phase 1: Foundation

Establish SSDF core practices and train the development team on secure coding.

2

Phase 2: Automation

Implement security scanning in CI/CD pipeline and automated testing.

3

Phase 3: Maturity

Refine processes based on metrics and expand security practices.

4

Phase 4: Continuous Improvement

Regular security assessments and framework updates as SnowBe grows.

Thank you