Software Engineering 2.0.0

BISHESH'S SURVEY 2022-04-01

Survey Summary

26 / 30	Quality and Health
13 / 15	Programming Concepts
20 / 24	Tooling
16 / 21	Integration
12 / 15	Full Stack
11 / 15	Vision and Growth
22 / 30	Architecture
120 / 150	Total

Quality and Health

Q1

K

Understands potential attack vectors within applications.

2 Familiar with the entire OWASP Top 10 for multiple application types.

References: OWASP • OWASP: Top 10 • OWASP: API Security

Q2

S

Writes code that protects against potential attack vectors.

3 Can write robust, and well-tested code to protect against many types of attacks for all layers of their team's applications. Makes security vulnerabilities a priority and proactively considers them during planning.

References: OWASP • OWASP: Top 10 • OWASP: API Security

Q3
B

Analyzes the team's applications attack surface and potential vulnerabilities.

2 Reviews both new code and old code for vulnerabilities and also actively refactors old code to make it more secure and resistant to attacks. Responds to security issues quickly.

References: OWASP • OWASP: Attacks • CVE

Q4
S

Understands the monitoring metrics and tools used by the company and how to integrate with them (e.g. logs, usage, system, security).

2 Can do complete setup for an application to connect to core services - either directly or through SDKs provided by the company.

Q5
S

Profiles an application to identify and implement performance enhancements.

3 Knows how to measure performance and bottlenecks at all tiers of their application and frequently optimizes application code to be more efficient. Considers performance tuning part of the shipping process.

Q6
K

Understands different metrics of application health and performance.

3 Understands the most important metrics for measuring application health and performance such as response time, throughput, error rates, logging and how to respond appropriately at various thresholds. Does what it takes to ensure a high-performance application (proactively and reactively).

Q7
B

Follows established development standards (security, performance, testing, coding style).

2 Follows company standards and understands the importance of adhering to them. Actively seeks out senior developers if unsure on best practices and when no clear company guidelines are available.

Q8
B

Analyzes the team's applications and fixes errors.

3 Proactively writes defensive code. Can quickly spot bugs or problematic code during code reviews. Has a deep understanding of the team's codebases and understands the importance of fixing bugs and minimizing Known Shippables.

Q9

Identifies, explains, and resolves software issues.

S

3 Can quickly troubleshoot issues, find bugs and their solutions. Efficiently identifies issues in pressure situations such as outages. The Go-To person when other developers cannot isolate and identify issues.

Q10

S

Writes code that fails gracefully and handles error conditions (e.g. service failures, logging).

3 Has deep knowledge and understanding of how error handling works in all aspects of the team's applications. Has written error handlers and knows how to safely log errors for monitoring. Writes code to protect against all situations - including rare failures or edge cases. Can write code that is defensive and fails gracefully for client or user experience.

Programming Concepts

C1

S

Writes maintainable and modular code.

3 Has written large-scale applications or libraries while maintaining solid software architecture using best practices. Writes thoughtful code that takes users into account and considers readability and comprehension a priority (principle of least surprise).

C2

K

Understands common development principles (e.g. DRY, KISS, YAGNI).

2 Has a complete understanding of several software development philosophies, and has experience using them in practice - including when ignoring them caused issues or made development more difficult.

References: Wikipedia: Software Development Philosophies

C3

K

Understands object-oriented programming paradigms and patterns.

3 Has in depth knowledge of most OOP design patterns and has built or managed large-scale software using them.

C4

K

Understands functional programming paradigms and patterns.

3 Has deep knowledge of most functional programming design patterns and has built or managed large-scale software using them.

C5

K

Understands different methods of encryption (e.g. symmetric, asymmetric, authenticated).

2 Understands the value of encryption and the importance of the various methods including symmetric, asymmetric, and hashing including when they should be used and what type of data they should be used with.

Tooling

T1

K

Understands the various types of testing and their importance (e.g. unit, functional, acceptance, user).

3 Has in-depth knowledge on the various types of testing and the value they add. Understands all types of testing and what is needed to make automated test suites robust enough to cover nearly every scenario and enable continuous deployment.

T2

B

Writes comprehensive tests for all code.

2 Has used both TDD and BDD and has significant experience with at least one of them. Helps push the bar higher on code coverage and other testing metrics for projects they are involved in.

T3

S

Performs VCS management such as committing, merging, handling conflicts.

3 Understands the meaning behind VCS commands and how they work. Comfortable with console or CLI. Can handle conflict resolution for any situation and recovering data when something goes wrong. Often helps others when they have a vcs-related issue.

T4

K

Understands best practices for multi-person or team workflows and branch/merging strategies.

2 Has an in-depth understanding of multiple development workflows such as git flow, trunk-based development, branch-based development and why some workflows may work better for certain situations. Involved in defining the workflow and improving the development and vcs integration process for their projects and team.

T5

K

Understands requirements and process for compiling and building applications in their domain.

3 Has extensive knowledge of how to build and compile projects across the enterprise for a wide range of project types - including awareness of both unix and windows-based projects. Actively works on process improvement for simpler, faster, and more reproducible builds. An expert in the release management process - often consults and assists other teams and developers with troubleshooting.

T6

K

Understands requirements and process for deploying applications in their domain.

3 Has extensive knowledge of how to deploy projects across the enterprise for a wide range of project types - including awareness of both unix and windows-based projects. Actively works on process improvement for simpler, faster, and more consistent deployments. An expert in the release management process - often consults and assists other teams and developers with troubleshooting.

T7

K

Understands the release management pipeline and how to set up for applications in their domain.

2 Knows the entire software development process and SDLC from task intake to production deployment including requirements gathering, testing and promotion through environments. Helps define the standards and processes that the team follows for release management.

References: Wikipedia: Release engineering

T8

K

Understands the implications of changing or updating dependencies.

2 Understands the implications of using third party code and evaluates when to use them based on security, reliability, and community support. Knows how to use native package managers for their team's applications such as nuget, composer, npm, hex, and others. Has knowledge of vendoring, locked dependencies or shrinkwrapping, and verifying trust such as signing code and authorship.

Integration

I1

K

Understands the value of frameworks/libraries and how to use them effectively.

2 Digs deeply into framework and library documentation or code and knows the frameworks and libraries used by the team's applications in great detail.

I2

K

Understands how to properly implement authentication and authorization.

2 Has experience with implementing both authorization and authentication for their team's applications. Understands the security requirements and concerns for gating access to sensitive data and follows best practices both from QL and the industry.

References: RBAC • ACL • Wikipedia: OpenID • Wikipedia: SAML

I3

B

Creates and publishes reusable libraries or modules.

3 Actively maintains large libraries or modules used across the company. Has great knowledge of maintaining packages and takes developer experience into account when making changes - including following consistent versioning schemes. Likely involved in the maintenance and administration of internal package repositories.

I4

B

Evangelizes and exhibits open-source mindset.

2 Understands the benefits QL receives from open-source and value of contributing back to the industry. Encourages collaboration through code reviews, knowledge share, and open collaboration on software projects.

References: ThoughtWorks: The Culture of Open Source

I5

K

Understands inter-system communication protocols (e.g. REST, SOAP, RPC, Queues).

2 Can build a service used by other services or systems using best practices and industry-standard protocols. Understands the value of following standards in designing public APIs.

References: Apache Thrift • Amazon Ion • Transit

I6

B

Respects how code changes may affect downstream consumers.

3 Takes care to present a thoughtful developer or client experience - such as simple APIs, pleasant documentation, and clear error messages. Has experience with or uses concepts such as API versioning, feature flags, semantic versioning in their systems or libraries to lessen change impact to consumers.

I7

B

Helps design and encourage community standards (e.g. performance, testing, coding style).

2 Understands the value of common standards and guides discussions and driving consensus among their team or multiple teams.

Full Stack

F1

S

Writes different types of applications (e.g. console, web, API, native, embedded).

3 Has architected or built a significant portion of several production-quality systems from multiple categories of application.

F2

K

Knowledgeable of multiple data management systems and when they are most effective.

2 Has great knowledge of at least one database system and used it extensively. Is familiar with other types of database systems. Has experience managing databases such as defining schema and writing queries, migrations, and backups.

F3

S

Can set up a complete local environment for a system (e.g. IDE, runtime, web servers, database).

3 Defines procedure and writes documentation for set up of development environment and has in-depth knowledge of system dependencies and how to install them - including any required VMs, databases or system tools. Actively works on process improvement and knowledge share with others.

F4

K

Understands underlying operating system and how to integrate and configure runtime for application code.

2 Knows how to install application runtimes from source if possible. Understands integration points between system and application - including dependencies on dynamically linked libraries or other system dependencies. Can install custom modules or extensions and knows to manage system dependencies through system package managers such as yum, apt-get, chocolatey, etc.

F5

S

Can make changes to all levels of the team's projects (e.g. client-side, frontend, backend, database).

2 Can make changes to all layers of the team's applications and actively improves their knowledge and familiarity with all components of the team's systems.

Vision and Growth

V1

S

Learns new libraries and how to use them effectively.

3 Can implement new libraries and frameworks and share that knowledge with others. Can dive deep into others' code to seek a greater understanding beyond what the documentation may provide. Uses third-party code wisely and avoids being tightly coupled to specific implementations. Knows how to analyze libraries for code quality and security.

V2

B

Avoids writing code unnecessarily by using existing internal or open source solutions.

2 Seeks out new libraries and technologies in their application space and how they make their applications better. Implements third party libraries when a high quality option exists and knows when it is valuable to write their own implementation that meets the needs of QL.

V3

S

Learns new languages or systems and knows when it is appropriate to introduce them.

2 Demonstrates curiosity and interest in learning new technology and applying it to QL to make themselves and others more efficient or produce higher quality software. Works with architects to investigate new technology or products when introduced into the company or the team's applications.

V4

B

Is actively involved with development communities (e.g. giving talks, helping others).

2 Gives talks during Hack Week and IT communities. Actively looks for ways to share knowledge and improve developer skills. Mentors other engineers in both an unofficial and official capacity.

V5

B

Stays current on new and upcoming technologies.

2 Shows active interest and curiosity in new initiatives and services or systems within QL. Stays up to date on new libraries or technology in their domain. Knows about upcoming changes or versions of libraries or frameworks used by the team and shares that knowledge with others.

Architecture

A1

B

Designs extensible and composable systems (e.g. services, APIs, modules).

3 Leads development for one or more large-scale extensible and modular systems. Drives best practices and engagement within their developer community for building better systems within the enterprise.

A2

K

Understands how their team's applications integrate with external systems.

2 Familiar with the dependency graph of their team's applications and what systems they are reliant on - including specific data contracts and specifications in addition to endpoints or services used. Understands service failures or errors and how to handle them gracefully with methods such as back-offs, exponential timeouts and when to surface those error conditions to consumers.

A3

K

Understands system architectural patterns and when to use them effectively.

2 Understands the benefits and trade-offs of different architectures and has experience using that knowledge to be a significant contributor to the design and implementation of a production system.

References: Wikipedia: Software Architecture Styles and Patterns

---

A4

K

Understands the flow of data used by their team's systems.

2 Has in-depth knowledge of methods of safeguarding data and follows best practices with regards to encryption and security. Understands the requirements and best practices for protecting data both internal and externally-accessible as well as at-rest and in-flight. Follows NFRs for data management.

A5

B

Addresses current and prevents future technical debt.

3 Carefully considers the impact of changes on their codebases and the future ability to make changes to it. Brings attention to technical debt and other intangible factors that may slow development - including presenting solutions to address them and prioritizing them. Designs and writes software that is more resilient to technical debt and actively researches for new patterns and behaviors to protect against it. Often has a technical vision for their projects and plans to address internal software technical debt as well as system or integration issues.

References: Wikipedia: Technical debt

A6

S

Can effectively explain architectural decisions to technical and non-technical partners.

2 Effectively describes in-depth software design and architecture designs to both technical and non-technical partners to drive consensus among projects and teams.

A7

B

Writes effective documentation both in and outside of code (e.g. comments, changelog, architecture, system design, gliffies).

2 Writes technical and non-technical documentation both in code and outside of code such as system references, project decisions and architecture. Keeps documentation up-to-date to ensure correctness and clarity.

A8

K

Has expert knowledge in a specialized domain.

2 Actively pursues their interest as a specialist in one or more significant business-impacting domains. Shows consistent progress and growth with their experience and knowledge.

References:

Wikipedia: Domain engineering • Wikipedia: Domain knowledge • Wikipedia: Domain

• Wikipedia: Subject-matter expert • ArtSoft: Domains of expertise

---

A9

K

Understands how to create scalable software.

2 Has in-depth knowledge of design patterns to improve scalability such as statelessness, caching, lazy loading, asynchronous processing and uses them in their software designs and implementations. Works with Ops to improve handling of availability, backups, monitoring, logging, and documentation.

A10

K

Understands how to create and manage distributed software and systems.

2 Has experience with setup, administration, and use of distributed systems such as distributed databases, clustered services, or peer-to-peer applications.