

# POLICIES AND PROCESSES ALL TECH STARTUPS NEED TO AVOID COSTLY MISTAKES



From a tech and development perspective, the early stage of a startup is both exciting and daunting. Laying a strong foundation from the outset can help you avoid costly mistakes and inefficiencies down the road.

Working in Chief Technology Officer roles for over 20 years, Tim Ng, Founder and Director of Scryla (your trusted tech expert) has developed the following list. We've put it into an infographic that will talk you through the key policies and processes all tech startups need within their tech departments.

Take a look at what you need.



## TWELVE CATEGORIES

At Scryla we believe that the policies and processes you need fall within the following twelve categories.

- ✓ Development workflow & processes
- ✓ Architecture & design
- ✓ Technology stack decisions
- ✓ Security
- ✓ Quality assurance
- ✓ Documentation
- ✓ Infrastructure & operations
- ✓ Collaboration & communication
- ✓ Skill development and training
- ✓ Vendor & tool management
- ✓ Performance & optimisation
- ✓ Feedback and iteration

### 1 DEVELOPMENT WORKFLOW & PROCESSES

#### Version control

Use tools like Git to track and manage code changes.

#### Code review

Establish a code review process to maintain code quality and foster collaboration.

#### CI/CD

Continuous Integration/Continuous Deployment (CI/CD) works best if you automate code testing and deployment to streamline development.

#### Environment management

Differentiate between development, staging, and production environments.

# 2

## ARCHITECTURE AND DESIGN

### System architecture

Design the technical architecture keeping scalability, maintainability, and performance in mind.

### Database design

Establish a normalised and scalable database schema and consider query optimisation.

### API strategy

Define conventions for API development, including RESTful practices, authentication, etc.

# 3

## TECHNOLOGY STACK DECISIONS

### Tech stack selection

Choose languages, frameworks, and tools that align with the startup's needs and team's expertise.

### Third party services

Evaluate and select external services/APIs for functionality like payments, messaging, etc.

# 4

## SECURITY

### Access control

Implement role-based access control for different parts of the system.

### Eryption

Encrypt sensitive data both at rest and in transit.

### Penetration testing

Regularly test the system for vulnerabilities.

### Security patching

Stay updated with patches for all software and dependencies.

# 5

## QUALITY ASSURANCE

### Testing strategy

Implement unit, integration, and end-to-end tests.

### Performance testing

Ensure that the system performs well under load.

### Bug tracking

Use tools like Jira or Bugzilla to track, prioritize, and manage bugs.



# 6

## DOCUMENTATION

### Code documentation

Comment code and maintain developer documentation.

### API documentation

Use tools like Swagger for API documentation.

### System documentation

Maintain documentation on system architecture, design decisions, and flowcharts.

# 7

## INFRASTRUCTURE AND OPERATIONS

### Backup and recovery

Regularly back up data and have a recovery strategy in place.

### Monitoring and alerts

Monitor system health and set up alerts for anomalies.

### Cloud strategy

Choose a cloud provider (e.g., AWS, Google Cloud, Azure) and define deployment strategies.

# 8

## COLLABORATION AND COMMUNICATION

### Agile development

Implement methodologies like Scrum or Kanban to foster iterative development and feedback.

### Documentation platforms

Tools like Confluence can be used for maintaining internal wikis and tech documentation.

### Communication tools

Use tools like Slack or Microsoft Teams for internal communication.

# 9

## SKILL DEVELOPMENT & TRAINING

### Training programs

Ensure the tech team is continually upskilling using training.

### Hackathons & innovation

Encourage creativity and new ideas through events or designated "innovation time."

# 10

## VENDOR AND TOOL MANAGEMENT

### Software licences

Manage and review licenses for software/tools to ensure compliance.

### Vendor evaluations

Periodically check third-party services for performance, cost, and security.

# 11

## PERFORMANCE AND OPTIMISATION

### Code profiling

Use profiling tools to identify performance bottlenecks.

### Optimisation

Regularly review and optimise code, database queries, and infrastructure.

# 12

## FEEDBACK AND ITERATION

### User feedback loop

Collect, review, and act on feedback from end-users.

### Post-mortem analysis

After major events or issues, run an assessment to understand what went wrong and how to avoid it in the future.

Join the conversation on LinkedIn  
> [scryla-consultancy-ltd](#)

### Contact us

Email: [info@scryla.co.uk](mailto:info@scryla.co.uk)  
[www.scryla.co.uk](http://www.scryla.co.uk)

