

# BLStream Fingerprint

# Table of Contents

1. Preface .....	1
2. Project management .....	2
2.1. Definition of Done .....	2
2.2. Daily Standup .....	2
2.3. Demo .....	2
2.4. Scrum/Kanban Board .....	2
2.5. Planning Meeting .....	2
2.6. Retrospectives .....	2
2.7. Retrospectives shared with the customer .....	2
2.8. Project Practices Charter shared with the customer .....	2
2.9. Collocated Team (all team members PM included) .....	2
2.10. 3rd party libraries licences listed, approved .....	2
2.11. Clean Backlog .....	2
2.12. Responsive Product Owner .....	2
2.13. Grooming .....	2
2.14. BurnUp / BurnDown chart .....	2
3. Development .....	3
3.1. Easy infrastructure setup .....	3
3.2. Easy application setup .....	3
3.3. Concurrency in application code accounted for .....	3
3.4. GUI Style Guide defined .....	3
3.5. Application Monitoring .....	3
3.6. Unit Tests .....	3
3.7. Scalability requirements known and accounted for .....	3
3.8. Performance requirements known and accounted for .....	3
3.9. Static code analysis (backend) .....	3
3.10. Application events logging .....	3
3.11. OWASP Top 10 in Definition of Done .....	3
3.12. Authorisation model defined .....	3
3.13. Continuous Integration .....	3
3.14. Continuous Delivery .....	4
3.15. Continuous Deployment .....	4
3.16. Documentation tracked in VCS .....	4
3.17. Documentation generated during CI .....	4
3.18. Parts of the documentation generated automatically .....	4
3.19. Automatic documentation of the executed tests .....	4
3.20. Documentation scope agreed .....	4
3.21. JS application framework .....	4
3.22. JS Build process .....	4
3.23. JS modules dependency management .....	4
3.24. JS Unit test .....	4
3.25. CSS builder .....	4

3.26. Static code analysis (Javascript).....	4
3.27. Truly REST-ful interfaces .....	4
3.28. HTML validator .....	4
3.29. Code Reviews .....	4
3.30. Pair Programming .....	5
3.31. Test Driven Development .....	5
3.32. Database schema versioning .....	5
3.33. Database data versioning .....	5
3.34. Concurrency for DB writes .....	5
3.35. Version Control System .....	5
3.36. Branching strategy .....	5
4. Quality assurance .....	6
4.1. Radiator .....	6
4.2. Defect Tracking System .....	6
4.3. Defined bug lifecycle .....	6
4.4. At least 1 QA for every 4 developers .....	6
4.5. Bug report template .....	6
4.6. Bug triage meeting .....	6
4.7. Smoke .....	6
4.8. Integration .....	6
4.9. Functional / Acceptance .....	6
4.10. Spelling .....	6
4.11. Security .....	6
4.12. Performance .....	6
4.13. Exploratory .....	6
4.14. Usability .....	6
4.15. Versioned repository of the test scenarios .....	6
4.16. Pair testing .....	6

# 1. Preface

BLStream Finger print is a set of practices applied in the company.

## **2. Project management**

**2.1. Definition of Done**

**2.2. Daily Standup**

**2.3. Demo**

**2.4. Scrum/Kanban Board**

**2.5. Planning Meeting**

**2.6. Retrospectives**

**2.7. Retrospectives shared with the customer**

**2.8. Project Practices Charter shared with the customer**

**2.9. Collocated Team (all team members PM included)**

**2.10. 3rd party libraries licences listed, approved**

**2.11. Clean Backlog**

**2.12. Responsive Product Owner**

**2.13. Grooming**

**2.14. BurnUp / BurnDown chart**

## **3. Development**

### **3.1. Easy infrastructure setup**

from nothing to running in <1h

### **3.2. Easy application setup**

from nothing to running in <1h

### **3.3. Concurrency in application code accounted for**

### **3.4. GUI Style Guide defined**

### **3.5. Application Monitoring**

### **3.6. Unit Tests**

Unit testing is at the core of engineering practices in BLStream. It's not just a practice, it is a foundation to many others, more sophisticated techniques like [Continuous Integration](#).

### **3.7. Scalability requirements known and accounted for**

### **3.8. Performance requirements known and accounted for**

### **3.9. Static code analysis (backend)**

### **3.10. Application events logging**

### **3.11. OWASP Top 10 in Definition of Done**

### **3.12. Authorisation model defined**

### **3.13. Continuous Integration**

- 3.14. Continuous Delivery**
- 3.15. Continuous Deployment**
- 3.16. Documentation tracked in VCS**
- 3.17. Documentation generated during CI**
- 3.18. Parts of the documentation generated automatically**
- 3.19. Automatic documentation of the executed tests**
- 3.20. Documentation scope agreed**
- 3.21. JS application framework**
- 3.22. JS Build process**
- 3.23. JS modules dependency management**
- 3.24. JS Unit test**
- 3.25. CSS builder**
- 3.26. Static code analysis (Javascript)**
- 3.27. Truly REST-ful interfaces**
- 3.28. HTML validator**
- 3.29. Code Reviews**

**3.30. Pair Programming**

**3.31. Test Driven Development**

**3.32. Database schema versioning**

**3.33. Database data versioning**

**3.34. Concurrency for DB writes**

**3.35. Version Control System**

**3.36. Branching strategy**



## **4. Quality assurance**

**4.1. Radiator**

**4.2. Defect Tracking System**

**4.3. Defined bug lifecycle**

**4.4. At least 1 QA for every 4 developers**

**4.5. Bug report template**

**4.6. Bug triage meeting**

**4.7. Smoke**

**4.8. Integration**

**4.9. Functional / Acceptance**

**4.10. Spelling**

**4.11. Security**

**4.12. Performance**

**4.13. Exploratory**

**4.14. Usability**

**4.15. Versioned repository of the test scenarios**

**4.16. Pair testing**