

Quality Improvement

Improve Quality of Bespoke Programming

- Skills Matrix
- ECRIN Standards

European Clinical Research Infrastructures Network (ECRIN)
(more detailed IT systems and Data Management specific standards)

Version 4.0, April 2018

<https://ecrin.org/sites/default/files/Data%20centre%20certification/Standards%20v4%20201804.pdf>

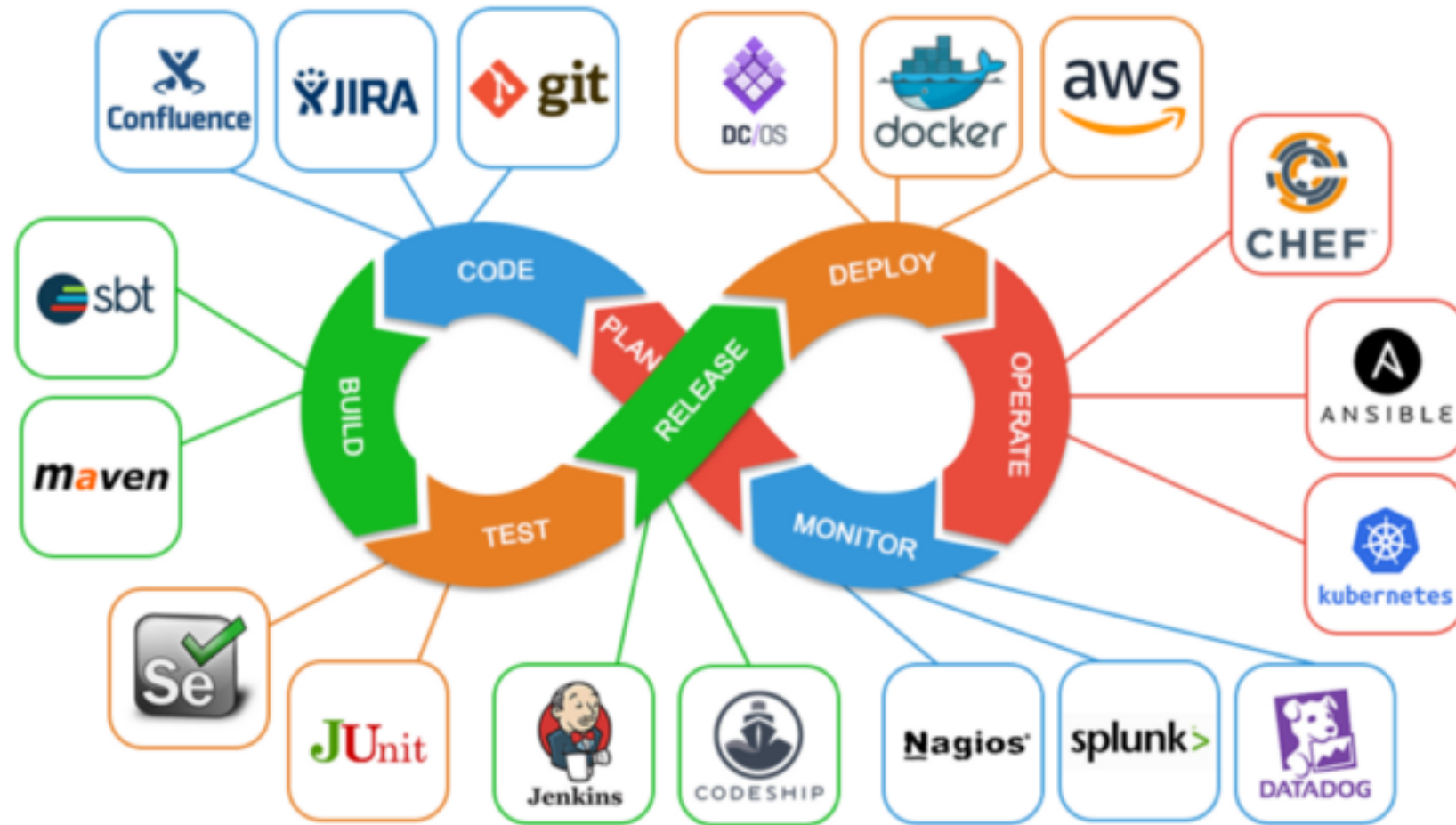
ECRIN Guidance for "Good practice in Software development"

- Design techniques that promoted clear 'separation of concerns' between different parts of a system.
- Use of a source control system that allows branching and release management.
- Programming against interfaces rather than concrete fixed components, with dependency injection.
- Programming against data repositories rather than fixed data sources.
- Use of a unit testing framework and / or integration tests.
- Continuous integration of a test regime with a source control system.
- Use of a library of user controls / common modules across systems.
- Regular code reviews and walk-throughs; shared coding.
- Use of a bug tracking system.
- Use of a scripted build and / or deployment scheme.
- Use of scripts for constructing and modifying databases.
- Consistent and effective error / exception handling techniques.
- Consistent and comprehensive logging techniques.

ECRIN Guidance Topics

| ECRIN Guidance | | Topic | |
|--|---|-------------------------|--------------------------------|
| Design techniques that promoted clear 'separation of concerns' between different parts of a system | 1 | Software Engineering | Methodology |
| Programming against interfaces rather than concrete fixed components, with dependency injection | | | |
| Consistent and effective error / exception handling techniques | | | |
| Use of a source control system that allows branching and release management | 2 | Version control | Methodology, Infrastructure |
| Programming against data repositories rather than fixed data sources | 3 | Data | Methodology, Infrastructure |
| Use of scripts for constructing and modifying databases | | | |
| Use of a unit testing framework and / or integration tests | 4 | Automated Testing | Methodology |
| Use of a library of user controls / common modules across systems | 5 | Lib Repository | Infrastructure |
| Regular code reviews and walk-throughs; shared coding | 6 | Code Review | Methodology |
| Use of a bug tracking system | 7 | Bug Tracker | Infrastructure |
| Use of a scripted build and / or deployment scheme | 8 | Build and Deployment | Infrastructure |
| Continuous integration of a test regime with a source control system | | | |
| Consistent and comprehensive logging techniques | 9 | Logging | Infrastructure |

Development Operations (DevOps)



ECRIN Guidance Topics + DevOps

| ECRIN Guidance | | Topic | | DevOps |
|--|---|----------------------|-----------------------------|-----------------|
| Design techniques that promoted clear 'separation of concerns' between different parts of a system | 1 | Software Engineering | Methodology | Plan |
| Programming against interfaces rather than concrete fixed components, with dependency injection | | | | |
| Consistent and effective error / exception handling techniques | | | | |
| Use of a library of user controls / common modules across systems | 2 | Lib Repository | Infrastructure | Code |
| Regular code reviews and walk-throughs; shared coding | 3 | Code Review | Methodology | Code |
| Use of a source control system that allows branching and release management | 4 | Version control | Methodology, Infrastructure | Code |
| Use of a unit testing framework and / or integration tests | 5 | Automated Testing | Methodology | Test |
| Use of a scripted build and / or deployment scheme | 6 | Build and Deployment | Infrastructure | Build & Release |
| Continuous integration of a test regime with a source control system | | | | |
| Programming against data repositories rather than fixed data sources | 7 | Data | Methodology, Infrastructure | Operate |
| Use of scripts for constructing and modifying databases | | | | |
| Consistent and comprehensive logging techniques | 8 | Logging | Infrastructure | Monitor |
| Use of a bug tracking system | 9 | Bug Tracker | Infrastructure | Monitor |

ECRIN Guidance Topics + DevOps + Progress

| ECRIN Guidance | | Topic | | DevOps |
|--|---|----------------------|-----------------------------|-----------------|
| Design techniques that promoted clear 'separation of concerns' between different parts of a system | 1 | Software Engineering | Methodology | Plan |
| Programming against interfaces rather than concrete fixed components, with dependency injection | | | | |
| Consistent and effective error / exception handling techniques | | | | |
| Use of a library of user controls / common modules across systems | 2 | Lib Repository | Infrastructure | Code |
| Regular code reviews and walk-throughs; shared coding | 3 | Code Review | Methodology | Code |
| Use of a source control system that allows branching and release management | 4 | Version control | Methodology, Infrastructure | Code |
| Use of a unit testing framework and / or integration tests | 5 | Automated Testing | Methodology | Test |
| Use of a scripted build and / or deployment scheme | 6 | Build and Deployment | Infrastructure | Build & Release |
| Continuous integration of a test regime with a source control system | | | | |
| Programming against data repositories rather than fixed data sources | 7 | Data | Methodology, Infrastructure | Operate |
| Use of scripts for constructing and modifying databases | | | | |
| Consistent and comprehensive logging techniques | 8 | Logging | Infrastructure | Monitor |
| Use of a bug tracking system | 9 | Bug Tracker | Infrastructure | Monitor |

Development Environment

Robust Test Harness

Auto Deployment

ECRIN Guidance Topics + DevOps + Progress + Expanded

| ECRIN Guidance | | Topic | | DevOps |
|--|---|----------------------|-----------------------------|-----------------|
| Design techniques that promoted clear 'separation of concerns' between different parts of a system | 1 | Software Engineering | Methodology | Plan |
| Programming against interfaces rather than concrete fixed components, with dependency injection | | | | |
| Consistent and effective error / exception handling techniques | | | | |
| Development Environment | | | Infrastructure | Code |
| <u>Standardized Development Environment</u> | | | | |
| Use of a library of user controls / common modules across systems | 2 | Lib Repository | Infrastructure | Code |
| Regular code reviews and walk-throughs; shared coding | 3 | Code Review | Methodology | Code |
| Use of a source control system that allows branching and release management | 4 | Version control | Methodology, Infrastructure | Code |
| Use of a unit testing framework and / or integration tests | 5 | Automated Testing | Methodology | Test |
| Robust Test Harness | | | | |
| <u>Automated User Acceptance Testing</u> | | | Methodology | Test |
| Use of a scripted build and / or deployment scheme | 6 | Build and Deployment | Infrastructure | Build & Release |
| Continuous integration of a test regime with a source control system | | | | |
| Programming against data repositories rather than fixed data sources | 7 | Data | Methodology, Infrastructure | Operate |
| Use of scripts for constructing and modifying databases | | | | |
| Consistent and comprehensive logging techniques | 8 | Logging | Infrastructure | Monitor |
| Use of a bug tracking system | 9 | Bug Tracker | Infrastructure | Monitor |