**Step by Step Guide to System Development in Technology Division**

**Roles:**

* DT - Development Team
* TT – Testing Team
* TPM – Technology Project Manager (DT + TT)
* THOD – Technology Head of Division (TPM+DT+TT)
* BT – Business Team
* BPM – Business Project Manager (BT)
* BHOD – Business Head of Division (BPM+BT)

**Steps:**

1. **Project Planning**
2. Project Concept (TPM + THOD)
3. Project Plan (TPM + THOD)
4. Project Progress Reports (TPM)

Project Closure Report (TPM) (step 26)

1. **Project agile development**
2. Agile software development process (scrum & sprint) (DT + BT as required)
3. Create a “Project Development Folder” (for user stories, sprint planning, product backlog, test planning etc.) (Lotus Notes / GitHUB / Taskboard) (DT + TT)
4. Software code - Write (SVN/GitHUB – Developer) (DT)

* **Source Control**

What tools are used in the CSO?

* + SVN is used for all internal projects
  + GitHUB is used for external public projects. An internal SVN repository may be needed for internal CSO customisations as part of an external project.

What is the recommended structure used?

* + Standard SVN structure is used: trunk, branch and tags

Access

* + To access SVN: raise a help desk ticket and ask for Admin access to VisualSVN
  + To access GitHUB: create an account on GitHUB*. << what else?>>*

Setting up a Project

* + Open VisualSVN, right click new project.
* **Software Development Tools:**
  + Front End Development / Client-Side Development install Microsoft Visual Studio Code

Required Extensions:

CSS Formatter

Document This

Live SAAS compiler

Toggle Format on Save

* + Server-Side C# Development install Microsoft Visual Studio Pro (2015) is the current version used

Required Extensions:

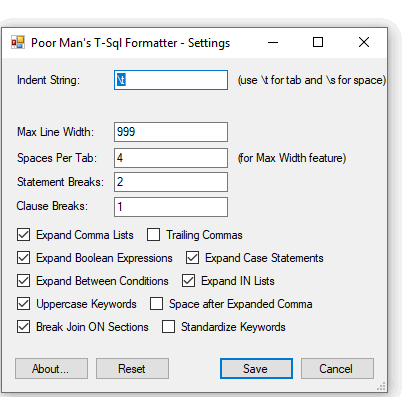
Format on Save

* + For SQL Server Database development install SQL Server Management Studio (10.7).

Required Extensions:

Poor Man SQL Formatter

Configure as follows:

: 

* **Long Term Monitoring**
  + API Level Logging

All events logged in one of the following three ways

* Append to a file: used for dev & testing
* Send via email: used for UAT & Prod
* Store on a db: always

Recommended Threshold Levels

* DEV: log all errors
* UAT: log all errors
* PROD: log fatal and critical errors only
  + Tracing

Logs every time an API is queried, who ran it and with what parameters.

1. Software code - Review (SVN/GitHUB - CoDeveloper) (DT)

* **Coding Standards**

<https://incubator.cso.ie/#>

* **Database Standards**

<<to be confirmed>>

|  |  |
| --- | --- |
| **Table names** | **prefixed with**:   * + td\_ (table with dynamic data)   + ts\_(table with static data\_   + tm\_(table with mapping data) |
| **Column Names** | **prefixed with**   * + 3 letter acronym representative of the table   + STATISTIC –> STT\_   + CLASSIFICATION -> CLS\_ |
| **Primary & Foreign Keys** | Single unique identifier (not a concatenated key)  No PK / FK in column name  PK – TBL\_ID  FK – TBL\_PKTBL\_ID  Examples: TD\_STATISTICAL; PK = STT\_ID  TD\_DATA: PK = TDT\_ID; FK = TDT\_STT\_ID |

1. Test - Plan (TT)
2. Deploy Test code into Test Server (DT)
   * + **Automated Testing & Deployment**

C#

At the moment automated testing is not implemented. Deployment follows SVN tagging

Client

No automated testing currently. Manual Export is sufficient

Database

Deployed using powershell

*<<reuse Lorenzo’s script – can we get a copy to link to>>*

1. Test - Execution (TT)
2. Bug Fixing (Developer / CoDeveloper) (DT)
3. Software sign off for UAT release (form) (TPM)

*(Please follow Technology Division Test Management Process)*

1. **UAT Release**
2. Tag UAT version in SVN/GitHUB (DT)
3. Deploy UAT Tagged version from SVN/GitHUB into UAT Server (DT)
4. Handover UAT release (including documentation) (DT to BT)
5. UAT and log bugs/feature requests in Bugzilla (BT)
6. UAT sign off with bugs (form) (BPM)
7. **Loop steps B-C till development completion**
8. **UAT Signoff**
9. UAT sign off without bugs, ready for Live Release (form) (BPM)
10. **Preparation for Live Release**
11. Prep for Backups, Pen-test, Load Test, Disaster Recovery, synch GitHUB (TPM)
12. Prep/Advance Communication (BPM)
13. Live Release sign off (form) (TPM + BPM)
14. **Live Release**
15. Live sign off (form) (TPM + BPM+THOD+BHOD)
16. Tag Live version in SVN/GitHUB (TPM)
17. Deploy Live Tagged version from SVN/GitHUB into Live Server (DT)
18. **Hypercare (short-term support)**
19. Vigilant Monitoring of Live Release (DT+BT)
20. Project Closure Report (TPM)
21. Bugs are logged through IT Service Desk/GitHUB and follow Steps B-H
22. **Maintenance (long-term support)**
23. Continuous Monitoring of the system (DT)
24. Bugs are logged through IT Service Desk/GitHUB and follow Steps B-H
25. **Change Requests**
26. Add requests into the “Project Development Folder” from IT Service Desk/GitHUB or annual requirements gathering (DT)
    * 1. A project plan is drawn up in Technology Division (TPM) where the effort required in Technology Division exceeds 20 person days effort. Steps A – H
      2. Small change requests below 20 person-days effort (DT+TT) follow Steps B-H
27. **Add Project to Bugzilla**
28. Development, Test & UAT bugs are tracked using Bugzilla.
    * + <https://bugzilla-incubator.cso.ie/>
      + Account: your test and development account.

Get added as an admin (Lorenzo can do this). Anyone else?

1. **Add Project to the Work Support System (WSS)**
2. Raise an IT Helpdesk ticket to get Admin access to the WSS
3. **Create Project Home Page on Lotus Notes**