



The ID Register

Software Development & Configuration Policy

Last Updated January 2017



Overview

The purpose of this policy is to manage software development and changes in a rational and predictable manner so that The ID Register staff and clients can plan accordingly. Changes require careful consideration, careful monitoring, and follow-up evaluation to reduce negative impact to users of The ID Register's services.

Change Management Policy

1. All changes to IT Services, except those that can be dealt with via a Standard Change for which operational procedures exist, are subject to this Change Management Policy and must follow the Change Management Process.
2. Only members of The ID Register's team can complete a Change Request form through TFS.
3. Only approved Changes can be scheduled for implementation.
4. A Change Advisory Board (CAB), appointed by the Director, will meet regularly to review change requests and to ensure that change reviews and communications are being satisfactorily performed.
5. A formal written Change Request must be submitted through the Change Portal for all changes, planned and unplanned.
6. All planned Change Requests must be submitted in accordance with Change Management process so that the CAB has time to review the request, determine and review potential failures, and make the decision to allow, delay or reject the request.
7. Planned change requests must receive formal CAB approval before implementation.
8. The appointed coordinator of the CAB may deny planned or unplanned changes for reasons including, but not limited to, inadequate planning, inadequate rollback plan, timing of the change will negatively impact a key business process, or if adequate resources cannot be readily available.
9. Staff notification must be completed for planned or unplanned changes following the steps contained in the Change Management Process.
10. A Post Implementation Review (PIR) must be completed for planned and unplanned changes, and whether successful or not.
11. Planned or unplanned changes to be implemented by external or 3rd party resources must be submitted by a member of The ID Register staff representative or IT service owner.

Software Development Policy

The ID Register and any associated software are developed using the Agile methodology.

1. The Business Analyst or “Product Owner” works closely with the team to identify and prioritize system functionality in form of a “Product Backlog” which is maintained in our TFS system.
2. The Product Backlog consists of features, bug fixes and non-functional requirements – whatever needs to be done in order to successfully deliver The ID Register.
3. With priorities driven by the Product Owner, the development team estimates and signs-up to deliver the next increment of the software during the next and successive Sprints, typically lasting 30 days.
4. Once a Sprint’s Product Backlog is committed, no additional functionality can be added to the Sprint except by the development team and Product Owner.
5. Once a Sprint has been delivered, the Product Backlog is analysed and reprioritized, if necessary, and the next set of functionality is selected for the next Sprint.
6. All code developed during this process is version controlled in TFS and developed exclusively in dedicated development environments for each developer.
7. Checked-In Code is automatically built into our Testing environment every day for review.
8. During the Sprint, the Testing environment is promoted into Acceptance periodically.
9. Each change in the Acceptance environment is downloaded from TFS and tested and documented.
10. The production software is promoted from Acceptance once a critical number of changes within the Sprint have been successfully tested.
11. Post Release delta testing is then completed and documented.

Open Source Components

The following open-source components are used within The ID Register.

- Angular - <https://angularjs.org/>
- Angular Cookie Law - MIT - <https://github.com/Palmabit-IT/angular-cookie-law>
- Angular Smart Table - MIT - <https://github.com/lorenzofox3/Smart-Table>
- Angular Tree Control - MIT - <http://wix.github.io/angular-tree-control>
- Angular UI Validate - MIT - <https://github.com/angular-ui/ui-validate>
- Angular Utils - MIT - <https://github.com/ngbp/ngbp>
- NG-Idle - MIT - <https://github.com/HackedByChinese/ng-idle>

- Bootstrap - MIT - <http://getbootstrap.com/>
- Bootstrap DateTime Picker - MIT - <https://github.com/Eonasdan/bootstrap-datetimepicker>
- Bootstrap File Input - <https://github.com/kartik-v/bootstrap-fileinput/blob/master/LICENSE.md> - <https://github.com/kartik-v/bootstrap-fileinput>
- Bootstrap Switch - MIT - <https://github.com/nostalgiaz/bootstrap-switch>
- Bootstrap Toggle - MIT - <https://github.com/minhur/bootstrap-toggle>
- Bootstrap Dialog - MIT - <https://github.com/nakupanda/bootstrap3-dialog>

- jQuery - <https://jquery.com/>
- Toastr - MIT - <https://github.com/CodeSeven/toastr>
- DataTables - MIT - <https://datatables.net>
- Handlebars - <https://github.com/wycats/handlebars.js/blob/master/LICENSE> - <https://github.com/wycats/handlebars.js/>
- Intl Tel Input - MIT - <https://github.com/Bluefieldscom/intl-tel-input.git>
- JQuery Password Strength - MIT - <https://github.com/ablanco/jquery.pwstrength.bootstrap>
- js-Cookie - MIT - <https://github.com/js-cookie/js-cookie>
- libphonenumber - Apache 2 - <https://github.com/googlei18n/libphonenumber>
- momentjs - <https://github.com/moment/moment/blob/develop/LICENSE> - <https://github.com/moment/moment/>
- Typeahead - <https://github.com/twitter/typeahead.js/blob/master/LICENSE> - <https://github.com/twitter/typeahead.js>
- Respond - MIT - <https://github.com/scottjehl/Respond>
- Fontawesome - GPL - <http://fontawesome.io/>

- EPPLUS - GNU - <http://epplus.codeplex.com/>
- Log4Net - Apache 2 - <http://logging.apache.org/log4net>
- Nettonsoft.Json - MIT - <https://github.com/JamesNK/Newtonsoft.Json>
- Ninject - Apache 2 - <http://www.ninject.org/>
- RestSharp - Apache 2 - <https://github.com/restsharp/RestSharp>
- Swashbuckle - BSD - <https://github.com/domaindrivendev/Swashbuckle>
- ReCaptureMVC - No Licence - <http://recaptchamvc.apphb.com>
- nClam - Apache 2 - <https://github.com/tekmaven/nClam>
- Moq - BSD - <https://github.com/Moq>
- Modernizr - MIT - <https://modernizr.com/>

Configuration Management

All configuration settings of the ID Register are managed by one of the two methods outlined below:

Server Settings:

Encryption keys and settings are stored outside the web application inside the Microsoft Azure Management Portal. Configuration files are version controlled within our TFS code development system and subject to our Change Management process.

High severity logs are reviewed periodically by the development team. Any changes are subject to our Change Management process.

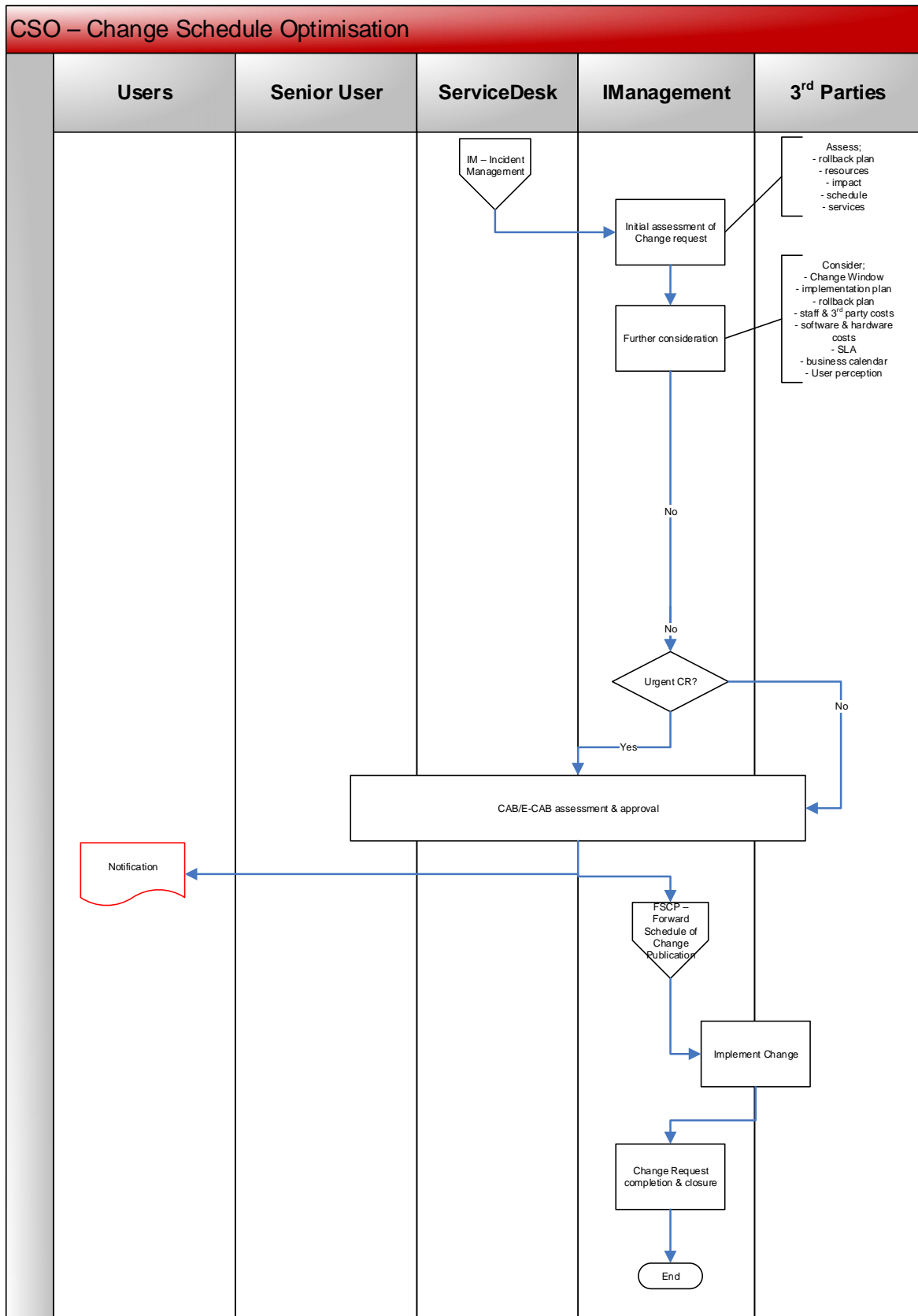
User Configuration:

All other configuration settings are maintained within The ID Register system in dedicated Administration screens. These screens are permissioned only to the following roles:

- Director
- Manager
- Business Analyst

Each administration change is audit logged and subject to our Change Management process as outlined above.

Appendix A – Change Schedule Optimisation



Appendix B – System Hosting Architecture

